

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка програмного забезпечення картографічного сервісу маршрутизації»

Студента 2 курсу, 5м групи,
спеціальності 121
«Інженерія програмного
забезпечення», спеціалізації
«Інженерія
програмного забезпечення»

підпис студента

Гелетуха Олександр
Аркадійович

Науковий керівник
кандидат технічних наук,
професор

підпис керівника

Рзаєва Світлана
Леонідівна

Гарант освітньої програми
доктор технічних наук,
професор

підпис керівника

Криворучко Олена
Володимирівна

АНОТАЦІЯ

Гелетуха О.А. Розробка програмного забезпечення картографічного сервісу маршрутизації

У випускній кваліфікаційній роботі розглянуто розробку картографічного сервісу маршрутизації, його компонентів та функціональності, а також використання JavaScript, та технології використання API MAPBOX GL JS для побудови кроссплатформених систем.

Ключові слова: Розробка програмного забезпечення, програмне забезпечення, API, JavaScript, MAPBOX.

ANNOTATION

Heletukha O.A. Rose of software protection of the cartographic routing service

The final qualification examines the development of a mapping routing service, its components and functionality, as well as the use of JavaScript, and the technology of using the MAPBOX GL JS API to build cross-platform systems.

Keywords: Software development, software, API, JavaScript, MAPBOX.

ТЕХНІЧНЕ ЗАВДАННЯ КАРТОГРАФІЧНОГО СЕРВІСУ МАРШРУТИЗАЦІЇ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Розробка програмного забезпечення картографічного сервісу маршрутизації.

Галузь застосування: Інформаційні технології, картографія.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на Випускню кваліфікаційну роботу, затверджене кафедрою програмної інженерії та кібербезпеки.

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Дана робота присвячена дослідженню проблем побудови маршрутів на графах реальних доріг та створенню мультиагентної системи для маршрутизації та створення оптимальних маршрутів. В роботі досліджується можливість створення оптимальних маршрутів з наявних картографічних даних, створюється граф доріг Києва, розглядаються існуючі методи пошуку оптимальних маршрутів.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Перед початком вибору технологій і проведення їх порівняльного аналізу необхідно виробити загальні вимоги до фінального виду програмної системи, які можуть кардинально вплинути на розглянуті технології. Для цього були обрані найважливіші, на думку автора, аспекти:

- кроссплатформенність - можливість запускати додаток на широкому спектрі пристроїв;
- платформи незалежність програмного коду - скорочення трудовитрат по підтримці нових платформ за рахунок використання існуючих технологій кінцевих середовищ виконання програм;
- клієнт-серверна реалізація - дистрибуція актуального програмного коду при кожному запуску програми, відмова від довгих оновлень і нових версій програми;
- повторне використання існуючого банку програмних API - скорочення трудовитрат по оновлення вже написаних, налагоджених і зарекомендувавши себе систем; деякі частини ПО буває простіше адаптувати під існуючу бібліотеку, ніж писати власну бібліотеку, пристосовану під додаток.

5. ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Технології що будуть застосовуватися при побудові:

- Мова програмування JavaScript;
- Текстовий редактор Visual Studio code для кодування;
- Технологія хосту сайтів та локальний сервер.

6. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання роботи повинна бути розроблена наступна документація:

- 1) Технічне завдання;
- 2) Програма та методика користувача;
- 3) Керівництво користувача.

7. ЕТАПИ ПРОЕКТУВАННЯ

- 1) Вивчення літератури за тематикою роботи;
- 2) Розробка та узгодження технічного завдання;
- 3) Розробка моделі функціонування;
- 4) Розробка структури картографічного сервісу маршрутизації;
- 5) Програмна реалізація картографічного сервісу маршрутизації;
- 6) Тестування картографічного сервісу маршрутизації;
- 7) Підготовка матеріалів текстової частини роботи;
- 8) Підготовка матеріалів графічної частини роботи.

Київський національний торговельно-економічний університет

Факультет ФОАІС Кафедра «Програмної інженерії та кібербезпеки»

Спеціальність 121 «Інженерія програмного забезпечення»

Спеціалізація «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри

_____ Криворучко О.В.

«___» _____ 2019 р.

Завдання

на випускн^у кваліфікаційну роботу студентів^і

Гелету^хи Олександр^у Аркадійович^у

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Розробка програмного забезпечення картографічного сервісу маршрутизації»

Затверджена наказом ректора від «07» листопада 2018 р. № 4183

2. Строк здачі студентом закінченої роботи (проекту) _____

3. Цільова установка та вихідні дані до роботи (проекту)

Мета роботи – полягає в розробці алгоритмів побудови персоналізованих навігаційних описів навігаційно-картографічних комп'ютерних систем для спрощення взаємодії людини з комп'ютером в області передачі інформації про шляхи і створенні програмних засобів на їх основі для використання в існуючих картографічних системах.

Об'єкт дослідження - розробка програмного забезпечення картографічного сервісу маршрутизації.

Предмет дослідження – побудова мультиагентної системи маршрутизації на основі алгоритмів пошуку маршруту.

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)
ВСТУП

**РОЗДІЛ 1. СУЧАСНІ МЕТОДИ ПОБУДОВИ І РОЗПІЗНАВАННЯ ОПИСУ
МАРШРУТИЗАЦІЇ ШЛЯХІВ**

- 1.1. Введення в предметну область
- 1.2. Аналіз існуючих навігаційних пристроїв і систем
- 1.3. Висновки до розділу 1

**РОЗДІЛ 2. ЗАПРОПОНОВАНІ МЕТОДИ НАВІГАЦІЙНО
КОМП'ЮТЕРНИХ СИСТЕМ**

- 2.1. Побудова текстового опису маршруту НККС
- 2.2. Розпізнавання текстового опису маршруту НККС
- 2.3. Висновки до розділу 2

РОЗДІЛ 3. РОЗРОБКА ТА ВПРОВАДЖЕННЯ НККС

- 3.1. Вимоги до програмної реалізації НККС
- 3.2. Архітектура програмного рішення НККС
- 3.3. Критерії вибору веб-фреймворків
- 3.4. Проблема адаптивного інтерфейсу користувача
- 3.5. Функціональні можливості НККС
- 3.6. Висновки до розділу 3.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	25.09.2018	
2.	<i>Розробка та затвердження завдання на проект магістра (Наказ №185 від 07.11.18)</i>	07.11.2018	
3.	<i>Вступ та перелік літературних джерел</i>	28.02.2019	
4.	<i>Розробка технічного завдання</i>	22.03.2019	
5.	<i>Розділ 1. Сучасні методи побудови і розпізнавання опису маршрутизації шляхів</i>	18.04.2019	
6.	<i>Розділ 2. Запропоновані методи навігаційно комп'ютерних систем</i>	24.05.2019	
7.	<i>Розділ 3. Розробка та впровадження НККС</i>	21.06.2019	
8.	<i>Розробка програми та методики тестування</i>	18.10.2019	
9.	<i>Написання наукової статті</i>	27.09.2019	
10.	<i>Керівництво користувача</i>	23.10.2019	
11.	<i>Висновки та пропозиції</i>	01.11.2019	
12.	<i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>	13.11.2019	
13.	<i>Підготовка автореферату та презентації доповіді</i>	14.11.2019	
14.	<i>Попередній захист випускної кваліфікаційної роботи</i>	14.11.2019 – 15.11.2019	
15.	<i>Здача зброшурованої випускної кваліфікаційної роботи</i>	25.11.2019	
16.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	26.11.2019	
17.	<i>Підготовка до публічного захисту випускної кваліфікаційної роботи</i>	02.12.2019- 04.12.2019	

7. Дата видачі завдання **«26» вересня 2018 р.**

8. Науковий керівник випускної кваліфікаційної роботи

(прізвище, ініціали, підпис)

9. Гарант освітньої програми

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент

(прізвище, ініціали, підпис)

11. Відгук керівника випускної кваліфікаційної роботи

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Науковий керівник випускної кваліфікаційної роботи _____
(підпис, дата)

Відмітка про попередній захист _____
(ПІБ, підпис, дата)

12. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента _____
(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми _____
(прізвище, ініціали, підпис)

Завідувач кафедри _____
(підпис, прізвище, ініціали)

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1	6
СУЧАСНІ МЕТОДИ ПОБУДОВИ І РОЗСПІЗНАВАННЯ ОПИСУ МАРШУТИЗАЦІЇ ШЛЯХІВ.....	6
1.1. Введення в предметну область	6
1.1.1 Електронна картографія та навігація	6
1.1.2 Підзадачі побудови і розпізнавання опису маршруту	8
1.1.3 Когнітивна навігація.....	9
1.2. Аналіз існуючих навігаційних пристроїв і систем	10
1.2.1 Автомобільні навігаційні пристрої.....	11
1.2.2 Мобільні навігаційні пристрої.....	12
1.2.3 Переносні навігаційні пристрої.....	13
1.3. Висновки до розділу 1.....	16
РОЗДІЛ 2.....	17
ЗАПРОПОНОВАНІ МЕТОДИ НАВІГАЦІЙНО КОМП'ЮТЕРНИХ СИСТЕМ.....	17
2.1. Побудова текстового опису маршруту НККС.....	17
2.1.1 Опис алгоритму навігаційно картографічних комп'ютерних систем.....	18
2.2 Розпізнавання текстового опису маршруту НККС.....	21
2.2.1 Опис алгоритму НККС	21
2.2.2 Проблема неточності введених користувачем відстаней	22
2.3. Висновки до розділу 2.....	24

					КНТЕУ 121 - 05-12.МР			
					Розробка програмного забезпечення картографічного сервісу маршрутизації	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		3	2	3
Зав. каф.	Криворучко О.В.							
Керівник	Рзасва С.В.							
Гарант	Криворучко О.В.							
Розробив	Гелетуха О.А.				Зміст	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

РОЗДІЛ 3	29
РОЗРОБКА ТА ВПРОВАДЖЕННЯ НККС	29
3.1. Вимоги до програмної реалізації НККС	29
3.2. Архітектура програмного рішення НККС.....	32
3.3. Критерії вибору веб-фреймворків	33
3.4. Проблема адаптивного інтерфейса користувача.....	34
3.5. Функціональні можливості	35
3.6. Висновки до розділу 3.....	36
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42

ВСТУП

Актуальність роботи – у сучасному світі інтеграція обчислювальної і роботизованою техніки в повсякденне життя стає помітнішою з кожним днем. Багато механічні та аналогові прилади замінюються електронними, набувають засоби для взаємодії з іншими пристроями, об'єднуються в мережі і утворюють так званий Інтернет речей. У цьому новому середовищі людині відводиться керуюча роль, він визначає завдання, які далі виконуються роботами, комп'ютерами та іншими електронними пристроями. Таким чином, в сферу спілкування людини, традиційно складається з його оточення, включаються і роботизовані системи. Підтримка з ними діалогу ускладнює специфіка самих пристроїв, їх стрімкий розвиток і необхідність постійного перенавчання. У зв'язку з цим на прикладному рівні виникає наукова проблема з розробки інтерфейсу взаємодії людини і роботизованих систем подібного роду.

Однією з таких проблемою є область передачі знання про маршрут, що може передаватися як від людини до роботизованою системі, так і в зворотному напрямку - від роботизованої системи до людини. Для передачі таких знань потрібен спеціальний формат опису маршруту, зручний для розуміння людиною і розпізнається роботом. Крім цього потрібно можливість персоналізації переданих даних під конкретного користувача. Її наявність дозволить скоротити опис, спростити його розуміння і збільшити ймовірність правильного розпізнавання при введенні. Наразі існує багато різних систем, які частково вирішують цю проблему. До таких систем можна віднести автомобільні навігатори, мобільні і стаціонарні платформи, веб-додатки та інші.

Мета дослідження – полягає в розробці алгоритмів побудови персоналізованих навігаційних описів навігаційно-картографічних комп'ютерних систем для спрощення взаємодії людини з комп'ютером в

області передачі інформації про шляхи і створенні програмних засобів на їх основі для використання в існуючих картографічних системах.

Предметом випускної кваліфікаційної роботи – розробка програмного забезпечення картографічного сервісу маршрутизації.

Об'єктом дослідження – побудова мультиагентної системи маршрутизації на основі алгоритмів пошуку маршруту.

Наукова новизна – полягає в розробці програмного забезпечення картографічного сервісу маршрутизації, з використанням сучасних мов програмування та технічної документації.

Методи дослідження – посібники з програмування та web-програмування, Інтернет ресурси пов'язані з програмуванням.

					<i>КНТЕУ 121 - 05-12.МР</i>	Арк
						4
Зм.	Лист	№ докум.	Підпис	Дата		

РОЗДІЛ 1

СУЧАСНІ МЕТОДИ ПОБУДОВИ І РОЗСПІЗНАВАННЯ ОПИСУ МАРШУТИЗАЦІЇ ШЛЯХІВ

1.1. Введення в предметну область

Сучасні навігаційно-картографічні системи мають багато різних форм і уявлень, починаючи з онлайнових сервісів, доступних на будь-якому комп'ютері, і закінчуючи мобільними пристроями, які завжди можна взяти з собою. Дана глава починається з введення в предметну область, містить основні визначення і терміни. Далі розглядаються основні підзадачі картографічних систем, серед яких виділяються побудова і розпізнавання опису маршруту. Наводиться порівняльний аналіз існуючих навігаційно-картографічних систем.

1.1.1 Електронна картографія та навігація

Тема електронної картографії та навігації дуже актуальна і затребувана на сьогоднішній день, цьому сприяло стрімкий розвиток:

- систем глобального позиціонування (GPS)
- телекомунікаційних технологій передачі даних на будь-які відстані;
- інформаційних банків даних картографічної інформації (Google, Яндекс та ін.).

Продуктом електронної картографії є навігаційно-картографічні системи - розподілені програмні комплекси з величезним масивом інформації, що зберігається в базах даних. Пристрої, на яких можна ними скористатися, дуже численні:

- персональні комп'ютери і ноутбуки;
- автомобільні навігатори;
- мобільні телефони, смартфони та планшети;
- смарт-годинник, смарт-окуляри;

					КНТЕУ 121 - 05-09.МР	Арк
Зм.	Лист	№ докум.	Підпис	Дата		6

- навігаційні стаціонарні термінал

Бази даних існуючих систем містять мільйони статичних об'єктів і величезна кількість динамічних даних по всьому світу:

- назви адміністративних і географічних об'єктів, назви доріг, адреси будинків, розмітка дорожнього руху, встановлений допустимий швидкісний режим;
- офіси організацій і компаній, соціальні установи, пам'ятки;
- фотографії, описи, коментарі користувачів, оцінки експертів;
- тимчасові події (наприклад, ярмарки);
- зупинки громадського транспорту, лінії його руху, розклад;
- поточний стан руху на дорогах (пробки, затори, ДТП);
- поточний стан окремих одиниць громадського транспорту.

Навігаційно-картографічні системи вирішують наступні завдання:

- надання актуальних статичних і динамічних даних;
- локалізація положення користувача;
- робота з маршрутами.

Робота з маршрутами має наступні підзадачі:

- автоматичний розрахунок оптимального маршруту між заданими точками його початку і кінця;
- побудова опису маршруту (передача інформації від системи до користувача);
- розпізнавання опису маршруту (передача інформації від користувача до системи);
- відстеження переміщення по заданому маршруту.

Інтерес для даної дипломної роботи представляють підзадачі побудови і розпізнавання опису маршруту. Вони мають готові рішення на сьогоднішній

					КНТЕУ 121 - 05-09.МР	Арк
Зм.	Лист	№ докум.	Підпис	Дата		7

день, однак ступінь їх зручності для користувача знаходиться на недостатньому рівні через накладених обмежень, розглянутих нижче.

В тексті буде використовуватися абревіатура POI, що позначає будь-яке місце інтересу (від англійського Point Of Interest): будівлі, організації, парки, пам'ятники мистецтва, пам'ятки, зупинки громадського транспорту і т.д. Терміни маршрут і шлях приймаються в якості синонімів.

У будь-якого шляху є точка початку (стартова) і кінця (фінішна). Маршрут є ламаною лінією, що представляє собою один з безлічі можливих варіантів і описує як дістатися з початкової точки в кінцеву, використовуючи один або кілька способів переміщення: пішки, на автомобілі, на літаку, на водному судні тощо. Між двома точками може бути будь-яку кількість можливих шляхів, в тому числі і нульове.

Ламана лінія складається з вершин (точок), з'єднаних прямими відрізками. Вершини ламаної лінії, а також точки початку і кінця маршруту мають географічні координати, що задаються довготою, широтою і Альтітуда (висота точки земної поверхні над рівнем океану), де це має сенс.

1.1.2 Підзадачі побудови і розпізнавання опису маршруту

Підзадача побудови опису маршруту полягає у створенні деякого уявлення автоматично знайденого шляху між наперед заданими точками його початку і кінця. Це подання використовується людиною і має бути:

- легко сприймаються - з метою скорочення витрачається користувачем часу для його розуміння;
- легко запам'ятовується - з метою використання користувачем отриманих від системи знань про маршрут без багаторазового звернення до неї;
- персоналізованих, тобто використовують знайомі для конкретного користувача уявлення і POI - з метою спрощення сприйняття і запам'ятовування;

- нечутливим до помилок - користувач може помилятися, наприклад, в відстанях, система повинна бути лояльна цього;

Вище описані загальні вимоги до рішень підзадач побудови і опису маршруту в навігаційно-картографічних системах, які використовуються для визначення конкретних вимог до відповідними алгоритмами, що розглядаються в даній дисертаційній роботі

Забігаючи трохи наперед, варто відразу визначити існуючі способи подання опису маршруту, доступні на сьогоднішній день. Незважаючи на те, що детальний аналіз методів викладено далі по тексту, на даному етапі необхідно їх визначити і дати назви. Таким чином, побудова опису маршруту між наперед заданими точками старту і фінішу зараз можливо двома варіантами:

- зображення траєкторії — вдає із себе вид зверху на карту з намальованою ламаною лінією руху;
- покрокова інструкція до — переміщення - послідовність дій (поворотів, розворотів і т.д.), виконання яких зможе привести від старту шляху до його фінішу.

1.1.3 Когнітивна навігація

В даній дипломній роботі введемо наступні визначення, які будуть матися на увазі по всьому тексту:

- Когнітивна карта (від лат. Cognitio - знання, пізнання) - спосіб знайомого просторового оточення. Когнітивні карти створюються і видозмінюються в результаті активної взаємодії суб'єкта з навколишнім світом. При цьому можуть формуватися когнітивні карти різного ступеня спільності, «масштабу» і організації (наприклад, карта-огляд або карта-шлях в залежності від повноти представленості просторових відносин і присутності вираженої точки відліку). Це - суб'єктивна

Зм.	Лист	№ докум.	Підпис	Дата

картина, що має насамперед просторові координати, в якій локалізовані окремі сприймаються предмети. Виділяють карту-шлях як послідовне представлення зв'язків між об'єктами по певному маршруту і карту-огляд як одночасне подання просторового розташування об'єктів. Спочатку термін був запропонований в 1948 в роботі американського психолога Е.Толмена «Когнітивні карти у щурів і людини».

- Навігація — процес управління деяким об'єктом в певному просторі пересування; підзадачею навігацією є маршрутизація - процес визначення маршруту проходження в деякій мережі оточення (графі перетинів доріг в місті, графі коридорів і сходів усередині будівель).
- Когнітивна навігація — процес управління деяким об'єктом в певному просторі пересування з урахуванням його персональних знань про оточення.

1.2. Аналіз існуючих навігаційних пристроїв і систем

В сучасному світі технології дуже швидко змінюються і розвиваються, тому необхідно уточнити, що аналіз проходив на переважно останніх версіях програмного забезпечення, доступних на момент написання. На жаль, номер збірки не завжди можливо визначити, особливо у онлайнових систем з безперервним висновком оновлень, з метою спрощення опису він опускається. Розгляду підлягали як спеціалізовані навігатори, переважно використовуються в автомобілях, так і спеціальні програми, що встановлюються на мобільні пристрої і перетворюють їх в аналоги спеціалізованих навігаторів. Також розгляду підлягали носяться пристрої (смарт-окуляри, смарт-годинник) і стаціонарні системи, що використовуються в великих будівлях або культурних центрах міст, часто відвідуваних туристами.

					<i>КНТЕУ 121 - 05-09.МР</i>	Арк
Зм.	Лист	№ докум.	Підпис	Дата		10

1.2.1 Автомобільні навігаційні пристрої

Автомобільні навігаційно-картографічні системи, іменовані в основному навігаторами, зараз дуже поширені, мають різні форм-фактори, проте завжди складаються з:

- графічного екрану, на якому відображається карта, місце перебування і прокладений маршрут;
- системи введення натисканням на екран або винесеними на корпус клавішами;
- динаміка, здатного відтворювати в достатній для сприйняття якості людський голос, який озвучує інструкції до переміщення.

Навігатори компанії Navitel (рис.1) мають безліч різних реалізацій, в тому числі для різних платформ. При цьому всі вони схожі по функціоналу і зовнішньому вигляду користувацького інтерфейсу. Система реалізує обидва варіанти опису маршруту: зображення траєкторії і покрокова інструкція до переміщення. Введення опису маршруту неможливий, є тільки функціонал для завдання стартовою і фінішною точок маршруту, а також уточнення його додатковими транзитними точками, що за своєю суттю розбиває маршрут на кілька послідовних шляхів.

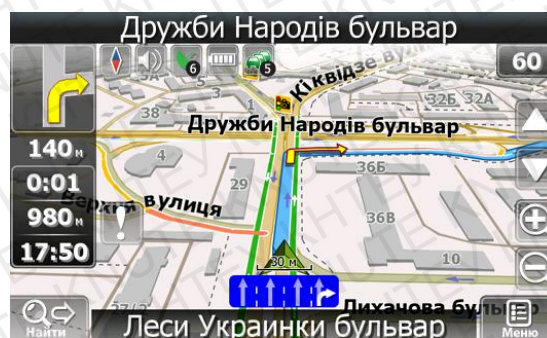


Рис.1: Приклад екрану системи Navitel

1.2.2 Мобільні навігаційні пристрої

Смартфони і планшети можуть вирішувати ті ж самі завдання, що й автомобільні навігатори. Для цього потрібна установка спеціального навігаційно-картографічного програмного додатка. Розгляду підлягали програми:

- Apple Maps
- Google Maps
- Яндекс Карти
- Яндекс Навігатор
- Mapbox

Всі розглянуті в даному розділі навігаційно-картографічні системи являють собою клієнтські програми зі зберіганням інформації на віддалених серверах. Таким чином, програмна система ділиться на дві частини:

- серверна частина:
 - зберігає інформацію про карти, зображеннях, POI тощо;
 - прокладає маршрути;
 - доступна по мережі Інтернет з будь-якої точки світу;
 - має постійно оновлювану актуалізується картографічну інформацію відповідно до змін в реальному світі.

Мобільна навігаційно-картографічна система Apple Maps від однойменної компанії Apple працює тільки на пристроях компанії, пропонує вже стандартні способи опису маршруту у вигляді зображення траєкторії (рис.2).

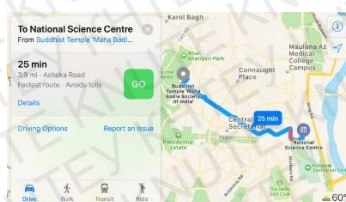


Рис.2: Приклад екрану системи Apple Maps

Зм.	Пис	№ докум.	Підпис	Дата

Навігаційно-картографічна система **Google Maps** від компанії Google реалізована для безлічі платформ, в тому числі для пристроїв фірми Apple і пристроїв на базі власної операційної системи Android. Функціонал мало відрізняється від розглянутої вище системи Apple Maps. Ті ж самі способи опису маршруту: візуальне уявлення траєкторії шляху і покрокова інструкція до переміщення без урахування персональних знань користувача про оточуючих об'єктах і історії його переміщень в минулому. Введення проміжних точок транзиту неможливий.

Система **Яндекс Карти** від компанії Яндекс широко поширена в основному на території Росії і СНГ. Додаток сконцентровано на надання довідкової інформації по розташуванню об'єктів і поточному стану пробок. Функціонал по формуванню опису прокладеного маршруту можливий тільки у вигляді візуального представлення траєкторії (рис.3). Введення маршруту можливий тільки через початкову та кінцеву точку без функціоналу щодо його уточнення.

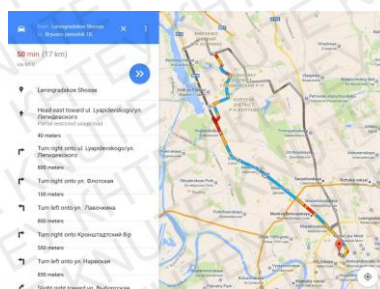


Рис.3 : Приклад екрану системи Google Maps (зображення траєкторії і покрокова інструкція до переміщення)

1.2.3 Переносні навігаційні пристрої

Смарт-годинник і смарт-окуляри стали проводитися порівняно недавно, але вже на поточний момент можуть допомагати користувачеві в навігації. Програмні рішення знаходяться в стадії глибокої розробки. Їх розгляд на поточний момент не відображає підсумкового якості продукту. Порівняно підлягали платформи:

- Google Wear
- Apple Watch
- Google Glass

Комп'ютерна система Google Wear (рис. 4) виробництва компанії Google працює в зв'язці зі смартфоном, передаючи системні дані по каналу Bluetooth. Навігаційно-картографічне додаток на смарт-годинах під'єднується до Google Maps і здійснює свого роду висновок на додатковий екран, доступний прямо на зап'ясті користувача, коли сам смартфон знаходиться в кишені або сумці. При запуску навігації до точки, заданої за допомогою голосової команди з використанням системи розпізнавання мови Google Now, на екрані смарт-годин відображається весь шлях цілком у вигляді зображення траєкторії і питається підтвердження користувача про те, що він дійсно бажає почати по ньому навігацію. Маршрут при цьому прокладається автоматично від поточного місця розташування.



Рис. 4: Приклад призначеного для користувача інтерфейсу системи Google Wear

Смарт-годинник Apple Watch (рис. 5) виробництва компанії Apple пропонують аналогічний функціонал. За початкову точку шляху приймається поточне місце розташування користувача, кінцева точка задається через голосового асистента Siri. Маршрут прокладається в автоматичному режимі без можливості його зміни. Перед початком навігації користувачеві показується шлях цілком у вигляді зображення траєкторії на карті місцевості.

Зм.	Лист	№ докум.	Підпис	Дата



Рис. 5: Приклад призначеного для користувача інтерфейсу системи Apple Watch

Аналогічним в плані навігаційно-картографічного функціоналу є смарт-окуляри Google Glass (рис. 6) від Google. Пристрій закріплюється на голові користувача як звичайні окуляри, з правого боку розташований маленький проектор, що проектує зображення на внутрішню сторону правого ока таким чином, що користувач як-будто бачить голограму призначеного для користувача інтерфейсу, що висить в повітрі на деякій відстані від особи. Додаток для здійснення навігації дуже схоже на згадані вище додатки для смарт-годин. Тим же самим способом, що і в Google Wear, користувач задає кінцеву точку маршруту за допомогою голосового помічника Google Now, за стартову точку береться поточне місце розташування, передане на окуляри зі смартфона, до якого вони приєднані по бездротовому каналу Bluetooth. Перед початком навігації на екрані відображається зображення траєкторії всього маршруту.

Всі розглянуті системи не мають можливості задавати той маршрут, який цікавить користувача. Тобто, можливий тільки автоматично прокладений шлях від поточного місця розташування до заданої голосом точки. Початкову точку поставити не представляється можливим. Кінцева точка задається, виходячи з заздалегідь доданих в «обрані» географічних координат, до яких поставлені у відповідність деякі назви, наприклад, «будинок» або «робота», або за допомогою голосового введення назви POI, це може бути ім'я ресторану

Зм.	Лист	№ докум.	Підпис	Дата

чи магазину, станція метро, назва вулиці або адреса цілком, який буде оброблений способом, аналогічним онлайнним навігаційно-картографічним системам, що пропонують смужку для текстового пошуку.



Рис. 6: Приклад призначеного для користувача інтерфейсу (того, що бачить користувач з надітим на голову пристроєм) системи Google Glass

1.3. Висновки до розділу 1

На основі проведеного в цьому розділі аналізу існуючих навігаційно-картографічних систем і застосовуваних в них способах побудови і розпізнавання опису маршрутів можна зробити наступні висновки:

- існуючі на сьогоднішній день в навігаційно-картографічних системах методи побудови опису маршруту не володіють одночасно зручністю для сприйняття і запам'ятовування, універсальністю використання на різних пристроях (обов'язково вимагають екран) і підстроюванням під знання конкретного користувача, котрі дають тим самим можливість здійснення когнітивної навігації;
- існуючі на сьогоднішній день в навігаційно-картографічних системах методи розпізнавання опису маршруту не універсальні (не можуть використовуватися на різних платформах, вимогливі до обладнання користувача) і не мають персоналізацією, тим самим не дозволяють використовувати описі знайомі конкретного користувача об'єкти і інтервали шляху.

Зм.	Лист	№ докум.	Підпис	Дата

РОЗДІЛ 2

ЗАПРОПОНОВАНІ МЕТОДИ НАВІГАЦІЙНО КОМП'ЮТЕРНИХ СИСТЕМ

2.1. Побудова текстового опису маршруту НККС

У цьому розділі розглядаються пропоновані методи побудови і розпізнавання текстового опису маршруту. Крім цього, зачіпаються проблеми персоналізації, чийм завданням є підстроювання поведінки алгоритмів під конкретного користувача і використання знайомих особисто йому місць і маршрутів на карті, проблеми спрощення одержуваного тексту з метою надання «людяності» формулювань і проблеми відміни назв.

Крім того, вирішуються питання, пов'язані з неточністю введеної користувачем інформації та наявністю декількох відповідних під опис маршрутів.

У загальному уявленні послідовність дій для побудови текстового опису маршруту складається з наступних кроків:

- отримання від користувача назв або координат точок старту і кінця шляху;
- побудова маршруту между заданими точками в автоматичності режимі з використанням існуючих навігаційно-картографічних онлайнових систем;
- отримання від навігаційно-картографічної системи докладного опису маршруту з конкретними директивами до переміщення;
- робота алгоритму побудови текстового опису на основі отриманих даних;

Всі пункти, крім 4-го, мають більше відношення до програмної реалізації, описаної в наступному розділі. Далі по тексту детально розібраний пункт четвертий.

В описі алгоритму часто наводяться приклади різних лексем російською мовою, які використовуються в фінальному текстовому поданні, наприклад, «вперед» або «півкілометра».

2.1.1 Опис алгоритму навігаційно картографічних комп'ютерних систем

Алгоритм базується на вхідних даних певного змісту, складається з послідовних кроків їх перетворення в підсумкове текстовий опис. Дані, одержувані на вхід від навігаційно-картографічної системи, являють собою трійку $In = \{L, M, D\}$:

- L — полігон маршруту;
 - являє собою масив двійок $\{Latitude, Longitude\}$ вершин ламаної лінії, яка описує траєкторію маршруту в реальному світі;
- M — текстові мітки з назвами вулиць;
 - являють собою масив трійок $\{Latitude, Longitude, Name\}$, в якому зберігається інформація про проїжджаємо користувачем вулицях;
- D — навігаційні директиви;
 - являють собою масив трійок $\{Latitude, Longitude, Directive\}$, в якому зберігається інформація про послідовних діях, які необхідно вжити для досягнення фінішної точки виду «прямо», «направо», «розворот» та інші.

Крім вхідних даних від навігаційно-картографічної системи використовуються додаткові джерела даних, що зберігаються локально:

- NL — знайомі користувачеві маршрути;
 - являє собою масив з іменованих масивів двійок $\{Latitude, Longitude\}$, що є знайомими користувачеві маршрутами або їх частинами, тобто тими інтервалами шляху, по яким він часто переміщається і може виділити для них особливу назву, наприклад, «далі, як на роботу»;
- NP — знайомі користувачеві POI;

- являє собою масив четвірок $\{Latitude, Longitude, Name, Popularity\}$, в якому зберігається інформація про місця, в яких часто бував користувач, в поле *Popularity* заноситься загальна кількість візитів;

Всі збережені локально дані мають власний унікальний ідентифікатор, який виражається цілим позитивним числом. Завдяки використанню повністю унікального ідентифікатора з'являється можливість використовувати адресацію до інформації по ньому без вказівки типу самих даних. Тут і далі ідентифікатор позначається як ID.

Високорівневе уявлення кроків роботи алгоритму:

- виділення з полігону маршруту, отриманого від навігаційно-картографічної системи, знайомих користувачеві інтервалів;
- фіксація прилеглих до маршруту знайомих користувачеві POI в якості орієнтирів;
- угруповання інтервалів по однотипним навігаційним директивам;
- формування каркаса результату з лексем;
- схилення назв і інтеграція їх в каркас;
- заміна залишилися лексем за словником
- формування результуючого текстового опису в форматі HTML з зображеннями і з візуальним форматуванням.

Угруповання інтервалів по однотипним навігаційним директивам відбувається наступним чином:

- всі директиви «прямо» групуються в одну, яку долає відстань підсумовується (відстань обчислюється виходячи з полігону, що описує конкретний інтервал шляху);
- всі директиви поворотів, відстань між якими менше заданого налаштування порога (передбачається, що це дуже близькі один до одного повороти), групуються в одну із зазначенням їх числа.

Подібні угруповання дозволяють скоротити підсумкове текстовий опис, тим самим спростивши його і зробивши більш легким для сприйняття і запам'ятовування.



Рис. 7: Пошук потрібної точки на шляху для отримання фотографії POI від навігаційно-картографічного сервісу

Для формування URL посилання необхідно обчислити:

- географічні координати точки з видом, звідки потрібно зображення POI;
- кут повороту камери.

Географічні координати POI заздалегідь відомі. Потрібно забрати позицію точки, з якої потрібно знімок, і кут повороту камери. Для цього потрібно виконати наступні дії:

- визначити найближчу точку кривої шляху до POI;
- відрахувати 100 метрів по ній в сторону старту;
- якщо це неможливо (POI знаходиться близько до початку інтервалу), то відрахувати 10% в сторону старту на частини інтервалу від його початку до її найближчої до POI точки;
- розрахувати кут повороту камери, виходячи з позиції, звідки робиться знімок, і географічних координат POI;

Результат формується в URL запит за правилами, визначеними навігаційно-картографічної системою.

Формування результуючого текстового опису в форматі HTML з зображеннями і візуальним форматуванням відбувається на останній стадії роботи алгоритму. Виконуються послідовно дії:

Зм.	Лист	№ докум.	Підпис	Дата

- всі лексеми, що містять назви POI ({POI NAME})
- спеціальний тег-блок, при наведенні на який (або натисканні на який при виведенні на сенсорний екран) показується спливаюче вікно з докладною інформацією про об'єкт:
 - офіційна назва POI;
 - текстовий опис зовнішнього вигляду;
 - зображення POI з того боку, з якою побачить його користувач.
 - підсумковий тест в форматі HTML виводиться на екран.

Візуально весь алгоритм побудови текстового опису маршруту може бути представлений таким чином у вигляді блок-схеми, зображеної на рис. 8.

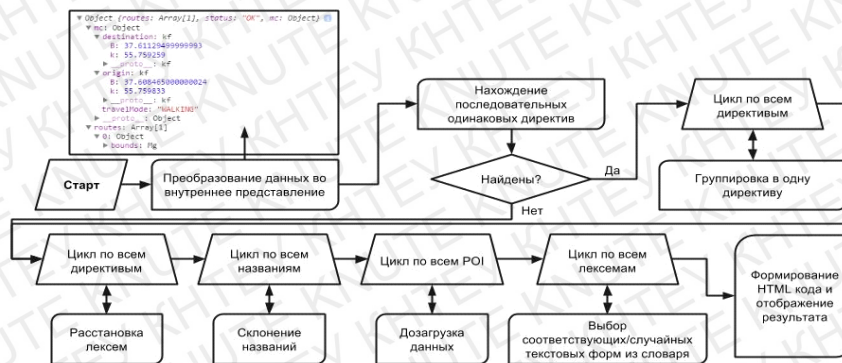


Рис. 8: Схема роботи всього алгоритму

2.2 Розпізнавання текстового опису маршруту НККС

Другим завданням алгоритму є розпізнавання текстового опису маршруту. Рішення являє собою, по суті, зворотний послідовність дій з декількома узагальненнями, що дозволяють скоротити обсяг введеної користувачем інформації.

2.2.1 Опис алгоритму НККС

- отримання призначеного для користувача введення, задає початкову і кінцеву точки маршруту:
 - користувач вводить тільки назви POI (вулиці, організації, об'єкти і т.д.), система в автоматичному режимі шукає підходящі і пропонує варіанти у спливаючому вікні або іншому зручному вигляді;

- отримання текстового опису шляху, що складається з навігаційних директив, назв POI, відстаней і навіть смітєвих слів:
- розбиття текстового опису на блоки четвірок $PB' = \{ACTION_TYPE, DIRECTION, POI_NAME, DISTANCE\}$, які можуть бути не повністю заповнені, де:
 - *ACTION_TYPE* — вид дії, яке може набувати значень виду «пройти» або «проїхати», що означатиме вид використовуваного транспорту (пішки або на автомобілі), що в свою чергу вплине на можливі варіанти прокладання шляху;
 - *DIRECTION* — напрямок руху, яке може бути описано в основному тільки словами виду «наліво» і «направо»;
 - *POI_NAME* — власне назва POI, поруч з яким відбувається дія;
 - *DISTANCE* — відстань в приблизному вигляді (очевидно, що користувач може помилятися), представляється зазвичай словами «кілометр», «чверть кілометра», «відразу» і іншими;
- прокладання можливих варіантів маршрутів між кожною парою POI і знаходження тих інтервалів, які підходять під залишилися заповнені поля четвірок PB' ;
- виставлення оцінки відповідності Λ_1 призначеному для користувача введення для кожної четвірки PB' і відкидання в разі, якщо вона менше Θ_1 ;
- об'єднання знайдених інтервалів в маршрути з виставленням оцінки відповідності Λ_2 призначеному для користувача введення;
- представлення всіх знайдених маршрутів з сумарною оцінкою Λ_2 , більшою деякого наперед заданого порогу Θ_2 .

2.2.2 Проблема неточності введених користувачем відстаней

Користувач, як будь-яка людина, може помилятися і вводити неточну інформацію. Для роботи алгоритму передбачається, що користувач точно вводить тільки назви POI (завдяки автоматичному доповнення тексту, яке виключає введення довільного неіснуючого назви POI), причому послідовність їх згадки правильна з точки зору їх появи на маршруті.

					КНТЕУ 121 - 05-12.МР	Арк
Зм.	Лист	№ докум.	Підпис	Дата		22

Для кожної четвірки PB' між двома заданими POI і для кожного знайденого між ними маршруту:

- перевіряється, що отримана оцінка відповідності Λ_1 більше наперед заданого значення Θ_1 , в іншому випадку відбувається відкидання знайденого інтервалу шляху;
- всі інтервали, які набрали задовольняє критерій оцінку, об'єднуються у відповідні маршрути, цілком заповнюють шлях від початкової до кінцевої точки;
- якщо такі маршрути вдалося отримати, то їм виставляється оцінка відповідності Λ_2 за правилами:
- оцінки всіх інтервалів підсумовуються;
- результат ділиться на кількість інтервалів, тим самим відбувається нормалізація оцінок;
- всі маршрути, які отримали оцінку відповідності Λ_2 вище наперед заданого значення Θ_2 , вважаються валідними і представляються користувачеві для вибору.

Користувач може вводити відстані неточно, помиляючись навіть на порядок. Передбачається, що фактично можна знайти маршрут між будь-якими двома заданими точками старту і кінця шляху. Для вирішення проблеми з відстанями застосовується наступна послідовність дій:

- якщо жоден із знайдених маршрутів не отримав оцінку відповідності Λ_2 вище Θ_2 , то вважається, що були невірно вказані відстані;
- проводиться повторний аналіз введених четвірок PB' без урахування поля *DISTANCE*, знайдені варіанти, якщо вони існують, представляються користувачеві з позначкою, що відстані не були враховані.

При порівнянні знайденого інтервалу з четвіркою PB' по полю *DISTANCE* відбувається перевірка відповідності відстані по наперед заданому критерію можливий відхилення Δ , зазначеним в процентному співвідношенні усіх відстаней.

Зм.	Лист	№ докум.	Підпис	Дата

2.3. Висновки до розділу 2

Інструментальні засоби мережевої безпеки є потужним захистом вільних мереж, навіть якщо користування передбачає звичайний web-серфінг. Якщо про загальний вид загроз як витік персональних даних чи деанонімізація в цілому, вже було наведено, то їх реалізація значно ширша та може включати як програмно-апаратні методи так і соціальну інженерію. Якщо другий варіант залежить тільки від дій самого користувача, то стати жертвою витоку даних через злам чи сніфінг значно складніше якщо подбати про персональну інтернет безпеку.

Програмні засоби та онлайн сервіси надають різні послуги, але мають спільну мету забезпечення захисту клієнтів. Небажане рекламне відслідковування лише найменша проблема. Сьогодні, здавалося б, звичайна рекламна аналітика може впливати на споживача не тільки заохочуючи до того чи іншого товару. На людях вже відпрацьовані онлайн механізми глобального впливу через такі системи, особливо під час політичних подій. В цифрову епоху на жаль залишаються і такі поняття як переслідування за цифровими даними, постами у соціальних мережах, цензура, тощо.

У розділі були проаналізовані існуючі засоби захисту користувача, такі як VPN, Proху та TOR-сервіси. Також було складено загальну модель захищеного інтернет-серфінгу. Захист інформації користувачів стала однією з галузей, для якої розробляються спеціальні інструментальні засоби, створюються вимоги до безпеки ІТ та оцінюється рівень їх виконання. Ефективність таких методів вже доведена. В деяких країнах доступ до всесвітньої мережі можливий саме засобами VPN та проксі, уникаючи переслідування та цензури. У вільних мережах, під час інтернет-серфінгу, це допомагає захистити користувача та його анонімність, чи вберегти персональне листування. Хоча існує досить розповсюджена думка що системи TOR, I2P тощо, використовуються переважно зловмисниками, саме такі надмережі є ідеальним захистом при

Зм.	Лист	№ докум.	Підпис	Дата

транзакціях криптовалюти, та анонімному спілкуванню. До того ж там можна знайти багато наукових робіт та вільних бібліотек.

Проблематика захисту буде підніматися частіше з розвитком новіших технологій. Вже зараз ідуть активні дослідження використання систем блокчейну, у космос запускаються супутники для покриття вільним інтернетом усієї планети а стандарти мобільного зв'язку змінюються на швидкісний 5G. Розвиток систем захисту повинен виконуватися прямо пропорційно науково-технічному прогресу, а розробка нового програмного забезпечення та методик захисту користувачів всесвітньої мережі є важливим завданням спеціалістів кібербезпеки. На щастя, на фоні світових подій, скандалів, повідомлень про витоки інформації чи навпаки, обмеження доступу, динаміка покращення безпеки рядових користувачів зростає.

					<i>КНТЕУ 121 - 05-12.МР</i>	Арк
						25
Зм.	Лист	№ докум.	Підпис	Дата		

РОЗДІЛ 3

РОЗРОБКА ТА ВПРОВАДЖЕННЯ НККС

3.1. Вимоги до програмної реалізації НККС

Перед початком вибору технологій і проведення їх порівняльного аналізу необхідно виробити загальні вимоги до фінального виду програмної системи, які можуть кардинально вплинути на розглянуті технології. Для цього були обрані найважливіші, на думку автора, аспекти:

- кроссплатформенність - можливість запускати додаток на широкому спектрі пристроїв;
- платформи незалежність програмного коду - скорочення трудовитрат по підтримці нових платформ за рахунок використання існуючих технологій кінцевих середовищ виконання програм;
- клієнт-серверна реалізація - дистрибуція актуального програмного коду при кожному запуску програми, відмова від довгих оновлень і нових версій програми;
- повторне використання існуючого банку програмних API - скорочення трудовитрат по оновлення вже написаних, налагоджених і зарекомендувавши себе систем; деякі частини ПО буває простіше адаптувати під існуючу бібліотеку, ніж писати власну бібліотеку, пристосовану під додаток.

Кроссплатформенність — у сучасній мережі пристроїв присутні як мобільні, так і стаціонарні обчислювальні системи. Першою вимогою до архітектури є максимальна підтримка всіх систем при найменших додаткових витратах на розробку під конкретну платформу. Даний пункт спирається на технології зі створення платформонезависимість програмного коду і не містить аналізів технологій. С# — програми, написані з використанням даної технології, не підтримуються жодної сучасної поширеною мобільною операційною системою, для їх запуску потрібне використання спеціальних проміжних бібліотек;

Зм.	Лист	№ докум.	Підпис	Дата

Клієнт-серверна реалізація — з метою спрощення передачі оновлених виконуваних файлів на різні платформи необхідно консолідувати їх на одному пристрої - сервері, передавальному їх через мережу Інтернет клієнтам (пристроїв, безпосередньо за яким працює користувач). Такий підхід дозволяє значно скоротити час розробки однієї ітерації і наступних оновлень вихідного коду. В якості клієнтів мається на увазі використання широкого набору різних платформ з індивідуальними архітектурою і відмінними операційними системами. Розглянемо з урахуванням цієї вимоги перераховані вище мови і використовуються спільно з ними технології:

- Java — програми, написані з використанням даної технології, зажадають перекомпіляції для роботи на мобільних пристроях сімейства Android, для платформ з операційною системою iOS потрібно використовувати додаткові програмні прошивки, що дозволяють запустити додаток; крім цього, повторна компіляція для різних платформ значно збільшить час однієї ітерації розробки, або звужить список підтримуваних пристроїв;
- C# — програми, написані з використанням даної технології, не підтримуються жодної сучасної поширеною мобільною операційною системою, для їх запуску потрібне використання спеціальних проміжних бібліотек;
- JavaScript — програми, написані з використанням даної технології, можуть бути запущені на будь-якій популярній і поширеній платформі всередині веб-браузера; для цієї мови не вимагається компіляція і можлива швидка ітерація шляхом запиту нового коду програми від сервера на кожному запуску програми;
- PHP — програми, написані з використанням даної технології, не можуть бути запущені без додаткових бібліотек на різних мобільних системах і вимагатимуть використання платформозавісімих API;
- Python — аналогічно PHP, програми не можуть бути запущені на різних мобільних системах.

Крім широкого спектру бібліотек, доступних для вільного використання, в наші дні стали популярні онлайн сервіси, що надають різноманітні безкоштовні послуги, що спрощують створення додатків. До таких відносяться:

- картографічні-навігаційні системи, що видають дані за маршрутами, POI тощо;
- хмарні сховища даних, що дозволяють отримати доступ до збереженої інформації з будь-якого пристрою;
- соціальні мережі, що дозволяють отримати список контактів і історію переміщень між POI;

Були розглянуті способи взаємодії з найвідомішими системами, які могли знадобитися для реалізації алгоритмів:

- Google Maps — навігаційно-картографічний сервіс;
- Bing Maps — навігаційно-картографічний сервіс;
- Яндекс Карты — навігаційно-картографічний сервіс;
- Карты Mail.Ru — навігаційно-картографічний сервіс;
- Dropbox — хмарне файлове сховище;
- Box — хмарне файлове сховище;
- Google Drive — хмарне файлове сховище;
- Облако Mail.Ru — хмарне файлове сховище;

Всі розглянуті системи підтримують систему REST запитів по протоколу HTTP / HTTPS в форматі JSONP. Все з розглянутих вище мов програмування підтримують роботу з даними форматом запитів.

Обрані технології — За підсумками розгляду технологій для реалізації була вибрана мова програмування JavaScript в його платформонезалежність інтерпретируемом поданні в поєднанні з технологіями HTML5 і мовою розмітки для користувача інтерфейсу HTML в зв'язці з CSS. На серверній стороні використаний сервер Apache. На клієнтській стороні роль інтерпретатора виконує веб-браузер.

Призначені для користувача дані зберігаються як локально в системі Storage веб-браузера, так і в хмарному сховищі. Доступ до API онлайн-сервісів

					КНТЕУ 121 - 05-12.МР	Арк
Зм.	Лист	№ докум.	Підпис	Дата		31

сервісів здійснюється по протоколу REST в форматі JSONP за технологією Ajax. Використовується підтримка безпечного зашифрованого протоколу передачі даних HTTPS з можливістю повернення до HTTP.

3.2. Архітектура програмного рішення НККС

За підсумками розгляду технологій в попередньому параграфі була розроблена архітектура, яка складається з таких компонент:

- сервер - призначений для зберігання вихідних кодів додатка з реалізацією алгоритмів на мові JavaScript і інтерфейсом на технології HTML5, переданих на клієнтські пристрої для виконання, при цьому сервер ніяких обчислень не виробляє;
- клієнт (и) - мобільні або стаціонарні комп'ютери, які мають встановлений веб-браузер переважно останньої версії (для повної сумісності і більшої ймовірності підтримки стандарту HTML5, прийнятого в поточному році, наприклад Firefox, Chrome, Opera, Safari), виконують отриманий від сервера код в автономному від сервера режимі, тобто передача даних від сервера до клієнта здійснюється тільки один раз в момент завантаження програми, всі інші дані виходять від онлайнових сервісів шляхом здійснення додаткових запитів.
- сервіси - доступні безкоштовні онлайнові сервіси, такі як: навігаційно-картографічні системи, хмарні сховища файлів, соціальні мережі.

Передача даних здійснюється по протоколах HTTP і HTTPS. HTTP, який не надає коштів шифрування, використовується для передачі вихідних кодів програми на клієнт з метою зменшення навантаження на сервер. HTTPS, що надає шифрування, використовується для передачі особистої інформації з соціальних мереж і для обміну файлами з хмарним сховищем.

Потоки передачі даних додатка сконцентровані навколо клієнтської сторони і мають орієнтовно наступну послідовність:

- З веб-браузера клієнта відбувається запит на сервер за певним URL адресою по протоколу HTTP методом GET;

- Сервер обробляє запит і відправляє клієнту index.html файл, який містить в собі розмітку призначеного для користувача інтерфейсу, посилання на файли з вихідними кодами на мові JavaScript і візуальними стилями оформлення на мові CSS3;
- Клієнт отримує index.html файл, робить додаткові запити на отримання файлів, зазначених в ньому у вигляді посилань, аналогічно по протоколу HTTP методом GET;
- Сервер відправляє запитані файли, на цьому спілкування сервера і клієнта закінчується;
- Клієнт відправляє запит в навігаційно-картографічну систему з даними по шляху, необхідними для роботи алгоритмів, по протоколу HTTPS методом GET, використовуючи технологію REST запитів;
- Сервер навігаційно-картографічної системи відповідає, передаючи необхідні дані клієнта;
- Можливі додаткові запити в соціальні мережі і хмарні файлові сховища аналогічно пунктам 5 і 6.

3.3. Критерії вибору веб-фреймворків

Для вирішення проблем, що виникають при реалізації, розумно скористатися існуючими фреймворками, головним завданням яких є спрощення та прискорення розробки. Для грамотного вибору відповідного програмного рішення були вироблені критерії:

- широка підтримка веб-браузерів;
- дозволяє працювати один раз написаному коду на максимально великому списку веб-браузерів, як підтримують стандарт HTML5, так і немає; обов'язковими для підтримки були Internet Explorer, Mozilla Firefox, Opera, Safari, Google Chrome (параметр версії веб-браузера на даній стадії розробки не є ключовим, передбачається, що користувач використовує останні версії програмного забезпечення);
- підтримка мобільних пристроїв;

Зм.	Лист	№ докум.	Підпис	Дата

- дозволяє працювати один раз написаному коду як на стаціонарних комп'ютерах, так і на мобільних без помітних відмінностей;
- відкритість вихідних кодів;
- дозволяє досліджувати технологію зсередини і в разі необхідності дописати потребується функціонал;
- вільна ліцензія;
- дозволяє вести безкоштовну некомерційну і комерційну розробку;
- наявність документації;
- дозволяє сконцентруватися на розробці власних вихідних кодів і використовувати технологію як «чорний ящик», що пропонує функціонал, описаний в документації замість дослідження його вихідних кодів.

Описані в даному розділі критерії використовуються далі по тексту при порівняльних оглядах існуючих технологій.

3.4. Проблема адаптивного інтерфейса користувача

Екрани мобільних і стаціонарних комп'ютерів мають різні розміри. Методом введення на стаціонарних комп'ютерах зазвичай є маніпулятор «миша», на мобільних платформах переважно використовуються тач-дисплеї, введення здійснюється безпосередньо дотиком потрібної області екрану. Обидві відмінні риси створюють проблеми при створенні користувацького інтерфейсу. Як і в інших ситуаціях стояло питання про створення коду UI тільки один раз, щоб він автоматично адаптувався під платформу. Для досягнення даної мети були досліджені існуючі фреймворки, націлені на вирішення цієї проблеми. Все фреймворки задовольняли обраними критеріями порівняння і мали лише незначні відмінності. Тому було прийнято рішення врахувати думку користувачів і порівняти їх відгуки на сайті GitHub, де зберігаються вихідні коди фреймворків.

3.5. Функціональні можливості

У підсумковій програмній реалізації є два функціональних компоненти: побудова текстового опису (рис. 9) і розпізнавання текстового опису (рис. 10). Приклади знімків екрану працюючої програми можна знайти в Додатку 1. Для побудови текстового опису використовується інтерфейс користувача з наступними задаються полями:

- початкова точка шляху;
- кінцева точка шляху.

Обидвоє поля мають автоматичні підказки, тобто користувач починає вводити Назву POI і відразу отримує список відповідних по назві. Далі, після підтвердження введення, в дію вступає алгоритм. Він отримує необхідну інформацію у онлайнової навігаційно-картографічної системи і будує текстове представлення в форматі HTML. На рис. 11 намальований приклад шляху, представлений у вигляді зображення траєкторії з мітками, використовуваними для його перекладу в текстове представлення.

Завдяки використанню HTML формату для подання текстового опису можливо:

- створення виділених жирним і курсивним шрифтом фрагментів тексту (такий стиль оформлення мають знайомі користувачеві POI або маршрути);
- створення спливаючих вікон з додатковою інформацією щодо POI (формальна назва, опис зовнішнього вигляду і зображення з тієї точки, з якої буде видно POI користувачеві в процесі переміщення по маршруту) при наведенні курсору миші на активний елемент або натискання на нього пальцем.

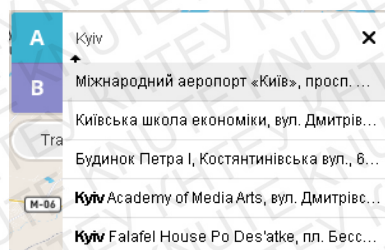


Рис. 9: Автоматичне доповнення назв POI

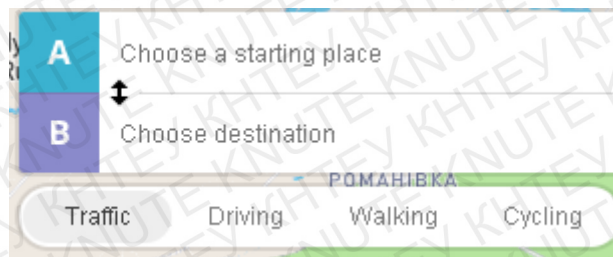


Рис. 10: Інтерфейс користувача для завдання маршруту та отримання текстового опису

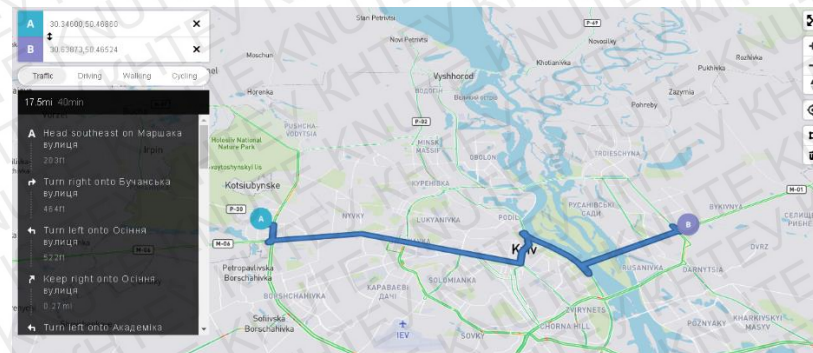


Рис. 11: Намальований приклад шляху

Користувач задає початкову і кінцеву точку маршруту, далі починає описувати маршрут текстом в спеціальному полі. При здійсненні введення спливають підказки з варіантами слів, які він може використовувати з метою скорочення числа помилок. У самому описі POI використовуються не безпосередньо, а через посилання на поля нижче, в яких вказуються точні назви POI за допомогою автоматичних підказок (аналогічно точкам початку і закінчення шляху).

3.6. Висновки до розділу 3.

У третьому розділі детально розглянута архітектура і реалізація програмного забезпечення, імплементує розроблені алгоритми побудови та розпізнавання текстового опису. Були розглянуті різні фреймворки і мови програмування, виявлено їх переваги та недоліки. У розділі докладно описана методика порівняння запропонованих алгоритмів з існуючими рішеннями з використанням методу Кано і приведена успішна статистика використання розроблених алгоритмів на практиці. На основі зібраних даних можна зробити висновок про затребуваність описаної розробки зокрема і самого завдання з побудови та розпізнаванню описів маршрутів в загальному.

Зм.	Лист	№ докум.	Підпис	Дата

Зм.	Лист	№ докум.	Підпис	Дата

КНТЕУ 121 - 05-12.МР

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

На основі проведеного дослідження методів введення і виведення маршруту розроблені і реалізовані нові алгоритми побудови та розпізнавання персоналізованого когнітивного текстового опису маршруту, що спрощують використання навігаційних систем.

На основі проведених досліджень і експериментів з формування когнітивної карти в свідомості людини і сприйняття текстового опису шляху розроблені алгоритми для зберігання і відображення даних про маршрут, що враховують психологію сприйняття картографічної інформації. Запропоновано методику перевірки якості пропонованих текстових описів і їх зручності для людини на основі статистики використання та анкетування, що виявляє об'єктивну думку користувачів.

На основі розроблених методів і алгоритмів реалізована програмна система, інтегрована в існуючі картографічні системи як модуля і використовується в реальних практичних додатках.

Розроблено та реалізовано нові алгоритми побудови та розпізнавання персоналізованого когнітивного текстового опису маршруту, що спрощують використання навігаційно-картографічних систем. Запропоновані алгоритми інтегрованими в широкий клас навігаційних систем, що працюють як на стаціонарних, так і на мобільних платформах.

Алгоритм побудови текстового опису враховує персональні знання користувача про оточуючих об'єктах, його пересування в минулому, часто відвідувані місця і іншу інформацію, що дозволяє скоротити і спростити опис.

					<i>КНТЕУ 121 - 05-12.МР</i>			
					<i>Розробка та впровадження веб-сервісу захищеного інтернет-сервісу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Арку</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>ВП</i>	<i>39</i>	<i>44</i>
Зав. каф.	Криворучко О.В.							
Керівник	Лашорін В.І.							
Гарант	Криворучко О.В.							
Розробив	Залевський Б.П.				<i>Висновки та пропозиції</i>	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

Алгоритм розпізнавання текстового опису підбирає відповідні під введенний текст маршрути з урахуванням даних, накопичених про користувача раніше, що дає можливість значно зменшити число помилкових варіантів. У разі якщо інформації недостатньо для точного визначення маршруту, то користувач може його уточнити додатковим введенням.

Для побудови текстових описів розроблений допоміжний алгоритм синтезу пропозицій, заснований на імовірнісних величинах, що надає вид живої мови формованому тексту.

На основі проведених досліджень і експериментів з формування когнітивної карти в свідомості людини і сприйняття текстового опису шляху запропоновані нові алгоритми для зберігання і відображення даних про маршрут, що враховують психологію сприйняття картографічної інформації. Завдяки такому підходу можливо спрощення сприйняття користувачем інформації про шляхи, а також зменшення обсягу необхідної інформації для здійснення навігації, що зберігається на пристрої.

					<i>КНТЕУ 121 - 05-12.МР</i>			
					<i>Розробка та впровадження веб-сервісу захищеного інтернет-сервісу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Арку</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>ВП</i>	<i>39</i>	<i>44</i>
Зав. каф.	Криворучко О.В.							
Керівник	Лашорін В.І.							
Гарант	Криворучко О.В.							
Розробив	Залевський Б.П.				<i>Висновки та пропозиції</i>	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

Перспективи подальшої розробки мають полягати в динамічних системах захисту, з використанням технологій нейромереж та штучного інтелекту, наприклад динамічне перемикання серверів залежно від навантажень, чи системи автоматичного схову користувача, при загрозі небажаного спостереження. Використання захищених проксі-серверів чи VPN мереж мають стати стандартом при користуванні інтернетом навіть під час домашнього користування. Представники міжнародних організацій та уряду мають більш глибоко вивчати цю проблему, та бути готовими до негативних подій у кіберпросторі, і зосереджувати особливу увагу на допомозі провідним інженерам програмного забезпечення для побудови комплексних систем захисту мільярдів користувачів всесвітньої мережі.

Зм.	Лист	№ докум.	Підпис	Дата

КНТЕУ 121 - 05-12.МР

Арк

40

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Статті/тези, опубліковані у збірниках наукових праць/конференцій.

1. Геллетуха О.А., USING KALI LINUX IN THE EDUCATIONAL PROCESS: матеріали Всеукраїнської науково-практичної конференції Безпека соціально-економічних процесів в кіберпросторі / Залевський Б.П., Пашорін В.І. – К.: Київ. КНТЕУ, 2019 – с. 226-227

Електронні ресурси.

1. YouTube - Алгоритм поиска A* [Електронний ресурс]. – Режим доступу: <https://www.youtube.com/watch?v=AsEC2TJZ3JY>
2. Coursera - A* Algorithm [Електронний ресурс]. – Режим доступу: <https://www.coursera.org/learn/algorithms-on-graphs/lecture/xXi6V/a-algorithm>.
10. E-maxx - Алгоритм Флойда-Уоршелла [Електронний ресурс]. – Режим доступу http://e-maxx.ru/algo/floyd_warshall_algorithm.
3. Ананий В. Алгоритмы: введение в разработку и анализ. / Ананий В., Левитин А. // В сб.: Introduction to The Design and Analysis of Aigorithms. — М.: «Вильямс», 2006. — с. 212-215.
4. Moore E. The shortest path through a maze. / E. Moore. - In.: Proceedings of the International Symposium on the Theory of Switching. — Harvard University Press, 1959. — pp. 285–292.
5. MJT - Contraction hierarchies path finding algorithm [Електронний ресурс]. – Режим доступу: <https://www.mjt.me.uk/posts/contraction-hierarchies/>.
6. aivanoff.blogspot - Агенты и мультиагентные системы [Електронний ресурс]. – Режим доступу: http://aivanoff.blogspot.com/2007/12/blog-post_18.html.

					<i>КНТЕУ 121 - 05-09.МР</i>			
					<i>Розробка та впровадження web-сервісу захищеного інтернет-сервісу</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Ар</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>СВД</i>	<i>41</i>	<i>44</i>
Зав. каф.	Криворучко О.В.							
Керівник	Рзасва С.В.							
Гарант	Криворучко О.В.							
Розробив	Гелетука О.А.				<i>Список використаних джерел</i>	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		

7. Мультиагентные системы [Электронный ресурс]. – Режим доступа: <https://intellect.ml/11-multiagentnye-sistemy-5354>.
8. Intellect - Мультиагентные системы [Электронный ресурс]. – Режим доступа: <https://intellect.ml/11-multiagentnye-sistemy-5354>.
9. Wooldridge M. An Introduction to MultiAgent Systems. / M. Wooldridge — Liverpool: John Wiley & Sons, 2002. — p. 366
10. Livejournal - Построение маршрута с помощью алгоритма Дейкстры [Электронный ресурс]. – Режим доступа: <https://efimmanevich.livejournal.com/732.html#comments>.
11. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. / [R. Geisberger, P. Sanders, D. Schultes, D. Delling, C. C. McGeoch]. // Springer, 2008. - Vol. 5038. – pp. 319–333.
12. Coursera - Contraction hierarchies – preprocessing [Электронный ресурс]. – Режим доступа: <https://www.coursera.org/learn/algorithms-ongraphs/lecture/AWFGa/preprocessing>.
13. Coursera - Contraction hierarchies – Witness search [Электронный ресурс]. – Режим доступа: <https://www.coursera.org/learn/algorithms-ongraphs/lecture/WuGFB/witness-search>.
14. Coursera - Contraction hierarchies Query [Электронный ресурс]. – Режим доступа: <https://www.coursera.org/learn/algorithms-ongraphs/lecture/XafED/query>.
15. Coursera - Contraction hierarchies – Proof of correctness [Электронный ресурс]. – Режим доступа: <https://www.coursera.org/learn/algorithms-ongraphs/lecture/uYeBL/proof-ofcorrectness>.
16. Coursera - Contraction hierarchies – Node ordering [Электронный ресурс]. – Режим доступа: <https://www.coursera.org/learn/algorithms-ongraphs/lecture/Abyzw/node-ordering>.
17. Microsoft Development Network – Windows Presentation Foundation [Электронный ресурс]. – Режим доступа: [https://msdn.microsoft.com/enus/library/ms754130\(v=vs.110\).aspx](https://msdn.microsoft.com/enus/library/ms754130(v=vs.110).aspx)

ДОДАТКИ

Додаток А

```
<!DOCTYPE html>

<html>

<head>

<meta charset='utf-8' />

<title>Display driving directions</title>

<meta name='viewport' content='initial-scale=1,maximum-scale=1,user-scalable=no' />

<script src='https://api.tiles.mapbox.com/mapbox-gl-js/v1.5.0/mapbox-gl.js'></script>

<link href='https://api.tiles.mapbox.com/mapbox-gl-js/v1.5.0/mapbox-gl.css' rel='stylesheet' />

<script src='https://api.tiles.mapbox.com/mapbox.js/plugins/turf/v3.0.11/turf.min.js'></script>

<script src='https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-draw/v1.0.9/mapbox-gl-draw.js'></script>

<link rel='stylesheet' href='https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-draw/v1.0.9/mapbox-gl-draw.css'

type='text/css' />

<link rel="stylesheet" href="js/mapbox-gl-traffic.js">

<link rel="stylesheet" href="css/mapbox-gl-traffic.css">

<link href="css/style.css" rel="stylesheet">

</head>

<body>

<script src='https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-directions/v4.0.2/mapbox-gl-directions.js'>

</script>

<link rel='stylesheet'
```

href='https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-directions/v4.0.2/mapbox-gl-directions.css'

type='text/css' />

<div id='map'>

</div>

</div>

<script src="[js/common.js](#)"></script>

</html>

```
mapboxgl.accessToken =
'pk.eyJ1IjoiYWxla3NhbmRya250ZXUiLCJhIjoiY2sycDN2NHdsMDBhbnlnZaXRobjdyaOSJ9
.kIUYSbFevwrNPC0DW2Iqlw';
var map = new mapboxgl.Map({
  container: 'map',
  style: 'mapbox://styles/mapbox/traffic-day-v2',
  center: [30.5, 50.43],
  zoom: 11,
  pitch: 45,
  bearing: -17.6,
  container: 'map',
  antialias: true
});
map.addControl(new MapboxDirections({
  accessToken: mapboxgl.accessToken
}), 'top-left');
map.addControl(new mapboxgl.FullscreenControl());
map.addControl(new mapboxgl.NavigationControl());
// map.addControl(new MapboxTraffic());
map.addControl(new mapboxgl.GeolocateControl({
  positionOptions: {
    enableHighAccuracy: true
  },
  trackUserLocation: true
}));
map.on('load', function () {
  // Insert the layer beneath any symbol layer.
  var layers = map.getStyle().layers;
  var labelLayerId;
  for (var i = 0; i < layers.length; i++) {
    if (layers[i].type === 'symbol' && layers[i].layout['text-field']) {
      labelLayerId = layers[i].id;
      break;
    }
  }
  map.addLayer({
    'id': '3d-buildings',
    'source': 'composite',
    'source-layer': 'building',
    'filter': ['==', 'extrude', 'true'],
    'type': 'fill-extrusion',
    'minzoom': 15,
    'paint': {
      'fill-extrusion-color': '#aaa',
```



```

        // use an 'interpolate' expression to add a smooth transition
        effect to the
        // buildings as the user zooms in
        'fill-extrusion-height': [
            "interpolate", ["linear"],
            ["zoom"],
            15, 0,
            15.05, ["get", "height"]
        ],
        'fill-extrusion-base': [
            "interpolate", ["linear"],
            ["zoom"],
            15, 0,
            15.05, ["get", "min_height"]
        ],
        'fill-extrusion-opacity': .6
    }, labelLayerId);
});

var draw = new MapboxDraw({
    displayControlsDefault: false,
    controls: {
        polygon: true,
        trash: true
    }
});
map.addControl(draw);

map.on('draw.create', updateArea);
map.on('draw.delete', updateArea);
map.on('draw.update', updateArea);

function updateArea(e) {
    var data = draw.getAll();
    var answer = document.getElementById('calculated-area');
    if (data.features.length > 0) {
        var area = turf.area(data);
        // restrict to area to 2 decimal points
        var rounded_area = Math.round(area * 100) / 100;
        answer.innerHTML = '<p><strong>' + rounded_area +
            '</strong></p><p>square meters</p>';
    } else {
        answer.innerHTML = '';
        if (e.type !== 'draw.delete') alert("Use the draw tools to draw a
        polygon!");
    }
}

```