Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА

на тему *«Метод оцінки відповідності маршрутів кораблів до еталонного маршруту»*

Галузь знань: *12 – Інформаційні технології* (Шифр і назва галузі знань)

Спеціальність: *122 – Комп'ютерні науки* (Шифр і назва спеціальності)

Освітня програма: *Комп'ютерні науки* (Назва освітньої програми)

Виконав:	*студент 4 курсу, група КН-18-1* (Курс, група виконавця)	Підпис	*Д.О. Боровик* (Ініціали, прізвище)
Керівник:	*д.т.н., завідувач кафедри КН* (Науковий ступінь, посада)	Підпис	*О.В. Бармак* (Ініціали, прізвище)
Нормоконтроль:	*доцент кафедри КН* (Науковий ступінь, посада)	Підпис	*Р.О. Багрій* (Ініціали, прізвище)

До захисту допускаю:
Зав. кафедри КН , д.т.н., професор — Підпис — *О.В. Бармак* (Ініціали, прізвище)

1 *червня* 2022 р.

Хмельницький 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет *інформаційних технологій*
Кафедра *комп'ютерних наук*
Освітній ступінь *бакалавр*
Галузь знань *12 – Інформаційні технології*
Спеціальність *122 – Комп'ютерні науки*

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*

_____
(підпис)
д.т.н., професор О.В. Бармак
«*25*» *березня* *2022* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: *Метод оцінки відповідності маршрутів кораблів до еталонного маршруту»*

2. Завдання видано студенту *Боровику Дмитру Олеговичу*
(прізвище, ім'я, по батькові)

3. Керівник роботи *професор кафедри КН Бармак Олександр Володимирович*
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «*1*» *березня* *2022* р. № *18*

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:
Мета роботи – опрацювання методу порівняння відповідності маршрутів суден відносно еталонного маршруту та розроробка інформаційної системи реалізації вказаного методу

Виконавець: *студент 4 курсу, група КН-18-1* _____ *Д.О. Боровик*
Курс, група виконавця — Підпис — Ініціали, прізвище

Керівник: *д.т.н., професор кафедри КН* _____ *О.В. Бармак*
Науковий ступінь, посада — Підпис — Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи: *«Метод оцінки відповідності маршрутів кораблів до еталонного маршруту»*

Виконавець кваліфікаційної роботи бакалавра: *студент групи КН-18-1 Боровик Дмитро Олегович*

Керівник кваліфікаційної роботи бакалавра: *завідувач кафедри КН Бармак Олександр Володимирович*

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
86	48	13	24	2

Метою кваліфікаційної роботи бакалавра є створення методу порівняння відповідності маршрутів кораблів відносно еталонного маршруту та розроробка інформаційної системи для реалізації вищезазначеного методу. Для виконання кваліфікаційної роботи було обрано мову програмування C# та платформу .NET Framework, фреймворк мови C# – ASP .Net. Для роботи із базою даних було обрано СКБД Microsoft SQL Server та мову запитів SQL.

Результатом виконання кваліфікаційної роботи є створений метод для порівняння маршрутів кораблів відносно еталонного маршруту та програмний продукт, призначений для оцінки маршрутів кораблів при переправах відносно еталонного маршруту, що може бути застосовано прикордонниками для підвищення ефективності виявлення порушників прикордонного законодавства. Програмний продукт розроблений на фреймворку ASP .Net із використанням СКБД – Microsoft SQL Server, мови програмування C# та мови запитів SQL.

Виконавець: *студент 4 курсу, група КН-18-1* (Курс, група виконавця) _____ (Підпис) *Д.О. Боровик* (Ініціали, прізвище)

Зміст

Перелік скорочень4

Вступ5

1 Характеристика предметної області і постановка задачі 7

1.1 Аналіз предметної області 7

1.2 Аналіз інформаційного забезпечення предметної області .....**Ошибка! Закладка не определена.**3

1.3 Постановка задачі..... 16

2 Проектування структури інформаційної системи 19

2.1 Аналіз та автоматизація обробки інформаційних потоків.....**Ошибка! Закладка не определена.**

2.1.1 Аналіз та створення методу для вирішення досліджуваної задачі.....**Ошибка! Закладка не определена.**

2.1.2 Аналіз питань щодо обробки інформаційних потоків 31

2.2 Розробка структури інформаційної системи 37

3 Програмна реалізація 46

3.1 Структура і функціональне призначення модулів системи, їх взаємозв’язок 46

3.2 Розробка програмних модулів 49

3.3 Тестування інформаційної системи 59

3.4 Інструкція користувача..... 65

3.5 Вимоги до публікації веб-застосування..... 67

Висновки 69

Перелік посилань.....**Ошибка! Закладка не определена.**

Додатки

Перелік скорочень

Скорочення, термін, позначення	Пояснення
АІС	Автоматизована інформаційна система
БД	База даних
ІТ	Інформаційні технології
КРБ	Кваліфікаційна робота бакалавра
КН	Комп’ютерні науки
ПЗ	Програмне забезпечення
ПП	Програмний продукт
СКБД	Система керування базами даних

Вступ

Інформаційна складова сучасної моделі охорони кордону на морській ділянці реалізується з використанням інтегрованої інформаційно-телекомунікаційної системи Морської охорони «Гарт-12». Застосування цієї системи дозволяє отримувати інформацію про поточне положення суден, порт відправки та порт призначення, тип вантажу та інші додаткові дані. При виборі судна у системі висвітлення надводної обстановки існує можливість перегляду попередніх точок його маршруту, отриманих у певні дискретні моменти часу. Останнє необхідне для встановлення можливих ознак порушення правил прикордонного режиму порушниками прикордонного законодавства. При цьому, візуалізація маршруту здійснюється з використанням лінійної апроксимації, а можливе порушення правил прикордонного режиму встановлюється на основі порівняння маршруту руху окремого судна та еталонного маршруту з кластера маршрутів, що з'єднують пункт відправлення та призначення судна. Однак, застосування лінійної апроксимації при формуванні неперервного апроксимаційного маршруту руху судна є сумнівним і нічим не підтвердженим.

Тому актуальним питанням є дослідження впливу виду апроксимації при побудові в системі висвітлення надводної обстановки неперервного маршруту руху судна через множину дискретних точок місцезнаходження судна у фіксовані моменти часу на величину метрики, що застосовується для встановлення подібності маршруту двох різних суден. Дослідження потребує: здійснення формалізації досліджуваної задачі; проведення аналізу можливості застосування для її вирішення лінійної, кусково-квадратичної апроксимації, апроксимації за допомогою інтерполяційного поліному Лагранжа, а також сплайн-інтерполяції; здійснення розрахунку метрики для встановлення ступеня подібності маршрутів суден; перевірку гіпотези про доцільність застосування лінійної апроксимації на основі порівняння отриманих результатів з використанням різних апроксимаційних методів. Актуальним є і питання

програмно-алгоритмічної реалізації вирішення досліджуваної задачі, що може сприяти автоматизації процесу перевірки гіпотези.

Отже, метою кваліфікаційної роботи бакалавра є опрацювання методу порівняння відповідності маршрутів суден відносно еталонного маршруту та розроробка інформаційної системи реалізації вказаного методу.

Об’єктом дослідження є маршрутизація руху суден в системах висвітлення надводної обстановки.

Предметом дослідження є методи встановлення відповідноності між маршрутами руху суден і еталонним маршрутом в системі висвітлення надводної обстановки.

1 Характеристика предметної області і постановка задачі

1.1 Аналіз предметної області

Специфічним і важливим видом національної безпеки є прикордонна безпека. Основні завдання в сфері прикордонної безпеки - забезпечення недоторканності державного кордону та охорони суверенних прав України в її виключній (морській) економічній зоні, покладаються на Державну прикордонну службу України (ДПСУ) [1]. Головні завдання, основні функції і принципи діяльності ДПСУ визначаються статтями 1-3 Закону України [1]. Основні напрями розвитку прикордонного відомства на найближчий період визначаються Стратегією розвитку [2] та Концепцією інтегрованого управління кордонами [3]. Згідно цих документів актуальними на даний час завданнями для ДПСУ вбачаються: забезпечення розвитку інтегрованого управління кордонами з урахуванням досвіду держав - членів Європейського Союзу; підвищення рівня боєздатності органів Держприкордонслужби та їх спроможності до виконання завдань із захисту державного кордону; забезпечення готовності до охорони тимчасово неконтрольованих ділянок державного кордону після відновлення контролю над ними; підвищення рівня довіри населення до ДПСУ та її особового складу. Реалізація цих завдань залежить від можливості вирішення ряду часткових завдань, окремими з яких є створення системи інтегрованого управління безпекою державного кордону та забезпечення розвитку Морської охорони Держприкордонслужби. Можливість вирішення останніх залежить від наявності та якості функціонування системи висвітлення надводної обстановки на морській (річковій) ділянці (СВНО). На це вказує той факт, що загальна протяжність державного кордону України становить 6992,982 км, з них загальна довжина морського кордону становить 1355 км. Тобто, майже 20% державного кордону України проходить по морю. Оскільки ж ефективність охорони морської (річкової) ділянки державного кордону залежить від якісного

функціонування СВНО, то це і підтверджує висновок про причинно-наслідкову залежність розвитку Морської охорони ДПСУ від розбудови відомчої СВНО.

Структуру та функціональні можливості вказаної СВНО можна оцінити з рис. 1.1.

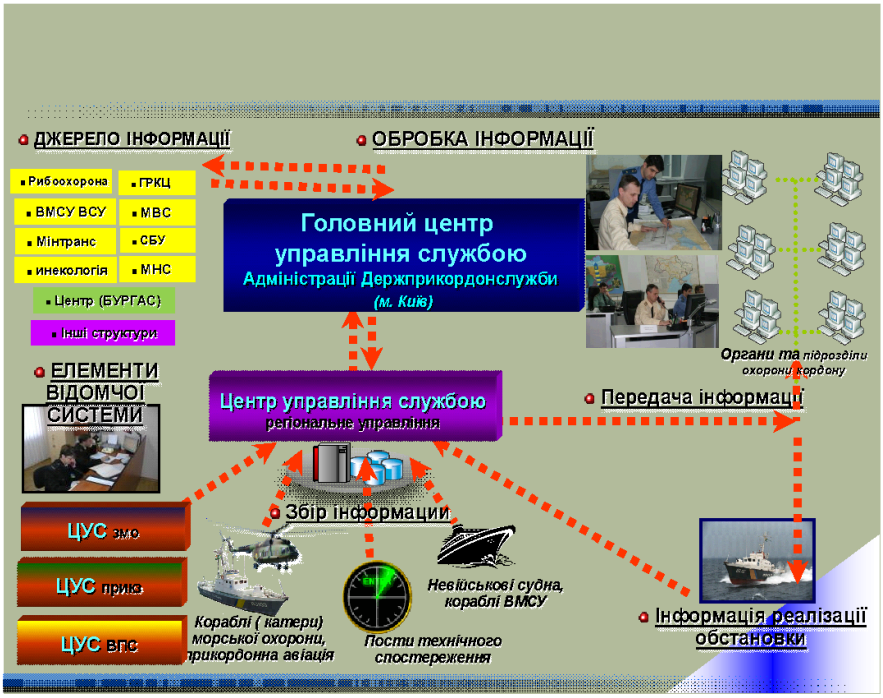


Рисунок 1.1 – Система висвітлення надводної обстановки ДПСУ

Питанням розбудови СВНО приділена увага в ряді робіт. Зокрема, в роботах [4-5] визначено методичні основи формування концепції розбудови СВНО на морській (річковій) ділянці в контексті забезпечення прикордонної безпеки та на основі цього безпосередньо сформовано концепцію. Функціональний аналіз можливих варіантів створення єдиної СВНО проведено у роботі [6]. У працях [7-8] визначено методичні основи оцінки ефективності функціонування СВНО на морській (річковій) ділянці та на основі цього сформовано методику оцінки такої ефективності.

Слід відмітити, що основу сучасної моделі охорони кордону на морській ділянці з використанням СВНО формує інформаційна складова. Остання реалізується з використанням інтегрованої інформаційно-телекомунікаційної системи (ІТС) Морської охорони «Гарт-12». Застосування ІТС «Гарт-12» дозволяє отримувати інформацію про поточне положення суден, порт відправки

та порт призначення, тип вантажу та інші додаткові дані. При виборі судна в СВНО існує можливість перегляду попередніх точок його маршруту, отриманих у певні дискретні моменти часу. Останнє необхідне для встановлення можливих ознак порушення правил прикордонного режиму. При цьому, візуалізація маршруту здійснюється з використанням лінійної апроксимації (наявні точки з'єднуються відрізками прямих), а можливе порушення правил прикордонного режиму встановлюється на основі співставляння маршруту руху окремого судна та кластера маршрутів, що з'єднують пункт відправлення та призначення судна.

На сьогодні ІТС «Гарт-12» ще не в повній мірі задовольняє потребам ДПСУ щодо виявлення потенційних і навіть реальних порушників прикордонного законодавства. Цим пояснюється той факт, що окремим напрямам удосконалення ІТС «Гарт-12» приділялася увага ряду науковців. Зокрема, визначенню підходів до просторового аналізу даних в ІТС «Гарт-12» приділена увага в роботі [9]. Формуванню методики виявлення просторових аномалій руху суден та її використанню при оцінюванні ризиків в ІТС «Гарт-12» присвячена праця [10]. У роботі [11] було обгрунтовано метод кластеризації маршрутів суден, а в роботі [12] - методику виявлення часових аномалій руху суден на основі даних ІТС «Гарт-12» у СВНО. Проведені дослідження дозволили опрацювати науково-методичні рекомендації органам управління прикордонного загону щодо побудови підсистеми технічного моніторингу надводної обстановки на морській ділянці відповідальності, які наведені в [13].

Разом з тим, на сьогодні ще залишається ряд невирішених питань, які не дозволяють в повній мірі ефективно застосовувати ІТС «Гарт-12» для виявлення порушників прикордонного законодавства на морській ділянці. До числа таких відносяться питання оцінки за допомогою діючої СВНО достовірності висновків щодо відповідності маршрутів кораблів еталонним маршрутам.

Підтвердженням коректності висновку про неефективність діючої ІТС «Гарт-12» є, зокрема, той факт, що застосування згаданої вище лінійної апроксимації при формуванні неперервного апроксимаційного маршруту руху судна, є сумнівним і нічим не підтвердженим [14]. Особливий сумнів виникає у

випадку значної віддаленості в часі або просторі досліджуваних точок, які характеризують положення судна.

Тому актуальним завданням, яке потребує аналізу і досліджуватиметься, в тому числі у даній роботі, є дослідження особливостей встановлення величини метрики, що застосовується для ідентифікації подібності маршрутів суден у СВНО. При цьому, потребуватиме уваги завдання оцінки впливу виду апроксимації при побудові в СВНО неперервного маршруту руху судна через множину дискретних точок місцезнаходження судна у фіксовані моменти часу на величину метрики, що застосовується для встановлення подібності маршруту двох різних суден. Дослідження передбачатиме: здійснення формалізації досліджуваної задачі; проведення аналізу можливості застосування для її вирішення лінійної, кусково-квадратичної апроксимації, апроксимації за допомогою інтерполяційного поліному Лагранжа та інтерполяційного поліному Ньютона, а також сплайн-інтерполяції; здійснення розрахунку метрики для встановлення ступеня подібності маршрутів суден; перевірку гіпотези про доцільність застосування лінійної апроксимації на основі порівняння отриманих результатів з використанням різних апроксимаційних методів.

Проведений аналіз дозволяє зробити висновок про те, що для ефективного використання ІТС «Гарт-12» у СВНО необхідно передбачити застосування програмного модуля, який би дозволяв формувати висновки про наявність фактів порушення прикордонного законодавства на морі на основі порівняння маршрутів кораблів з еталонними маршрутами. А для цього потребує розробки відповідна інформаційна система, яка б забезпечувала достовірність відповідних висновків на основі реалізації у ній відповідних видів апроксимації для формування неперервного апроксимаційного маршруту руху судна.

З урахуванням наведеного можна зробити висновок про те, що основними параметрами інформаційної системи для вирішення задачі, яка досліджується в даній роботі, є:

Параметр предметної області	**Опис**

Акваторія	
Тип акваторії	Назва типу акваторії
Назва акваторії	Географічна назва акваторії
Кодова назва акваторії	Назва акваторії, що застосовується в СВНО
Географічне положення акваторії	Географічні координати акваторії
Характеристика акваторії	Детальна інформація про акваторію
Тип акваторії	
Назва типу акваторії	Назва типу акваторії
Корабель	
Тип корабля	Назва типу корабля
Назва корабля	Офіційна назва корабля
Країна, якій належить корабель	Прапор, під яким ходить корабель
Кодова назва корабля	Назва корабля, що застосовується в СВНО
Номер корабля	Номер корабля у міжнародному реєстрі
Габарити корабля	Геометричні параметри корабля
Характеристики корабля	Детальна інформація про корабель
Тип корабля	
Назва типу корабля	Назва типу корабля
Еталонний маршрут	
Назва маршруту	Найменування маршруту
Тип корабля, для якого призначений маршрут	Назва типу корабля, для якого призначений маршрут
Назва акваторії	Географічна назва акваторії, у межах якої функціонує маршрут
Користувачі інформаційної системи	
Ім’я	Ім’я користувача
Прізвище	Прізвище користувача

Звання	Військове звання користувача (за наявності)
Посада	Посада, яку обіймає користувач
Дата народження	Дата народження користувача
Контактний телефон	Номер мобільного телефону користувача
Тип користувача	Назва типу користувача
Пароль	Таємне слово, яке слугує для авторизації користувача
Тип користувача	
Назва типу користувача	Прийнята для СВНО назва типу користувача
Спостереження	
Акваторія	Назва акваторії
Корабель	Кодова назва корабля
Еталонний маршрут	Назва маршруту
Дата спостереження	Дата та час спостереження
Статус	Висновки щодо необхідності реагування на рух корабля

1.2 Аналіз існуючого програмного забезпечення предметної області

На даний час інформаційні технології застосовуються практично в усіх структурах та організаціях для автоматизації більшості функціональних процесів, що реалізуються ними. Не становлять виключення і силові структури державних органів влади. Програмні продукти, що використовують ці структури зазвичай є надзвичайно надійними та безпечними. У процесі дослідження застосунків для автоматизації функціональних процесів, що реалізуються у силових структурах, у вільному доступі виявлено не було (у зв'язку з конфіденційністю останніх).

Разом з тим, інформаційну систему оцінки відповідності маршрутів кораблів еталонному маршруту можна віднести до класу експертних систем. Тому аналіз існуючого програмного забезпечення предметної області може

стосуватися огляду існуючих експертних систем, які є поширеними для застосування.

Експертна система – це методологія адаптації алгоритму успішних рішень однієї сфери науково-практичної діяльності в іншу. З поширенням комп'ютерних технологій це тотожна інтелектуальна комп'ютерна програма, що містить знання й аналітичні здібності одного чи кількох експертів в деякій галузі застосування і здатна робити логічні висновки на основі цих знань, тим самим забезпечуючи вирішення специфічних завдань без участі експерта [15].

Однією з ефективних на даний час експертних систем є система «PTV Visum», інтерфейс якої можна оцінити з рис. 1.2.

«PTV Visum» – експертна система та платформа для створення транспортних моделей міст, регіонів і цілих країн; інструмент оцінки і оптимізації громадського транспорту, обґрунтування проектів будівництва доріг, розв’язок і тунелів. Це програмне забезпечення є основою для розробки транспортних стратегій, комплексних схем транспорту, а також комплексних схем організації дорожнього руху [16].

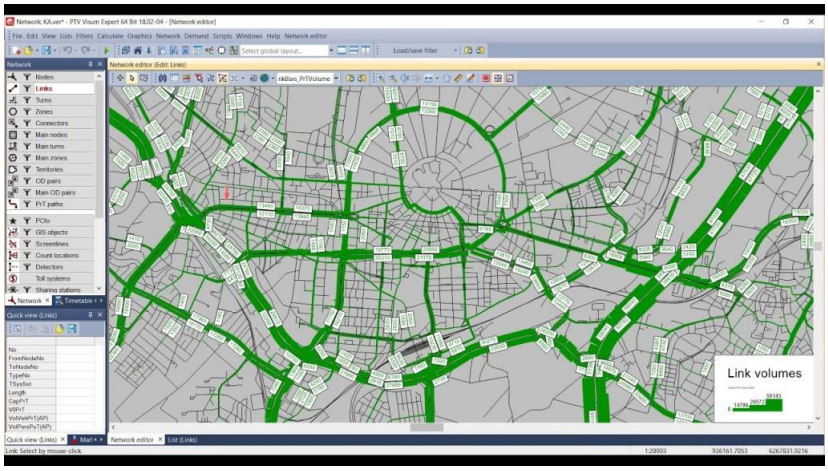


Рисунок 1.2 – Інтерфейс експертної системи «PTV Visum»

Основними перевагами цієї експертної системи є її можливості щодо:

– проведення аналізу дорожнього руху, прогнозу, керування даними на базі ГІС;

– врахування всіх учасників дорожнього руху, а також їх взаємодії;

– комплексного планування роботи громадського транспорту;

– розробки передових перспективних транспортних стратегій і рішень.

Проте зазначена система має і недоліки, що стосуються питань вартості та функціональної придатності. При цьому, останній недолік характерний для більшості експертних систем, оскільки, зазвичай, вони є зорієнтованими на вирішення конкретних задач.

Програмним продуктом, що призначений для вирішення окремих функціональних аспектів задачі оцінки відповідності маршрутів кораблів еталонному маршруту, є «Система оцінки подібності двох кривих». Дана система порівнює дві криві за одним із 5-ти способів побудови апроксимаційної кривої по дискретних точках, що належать їй. Система в результаті роботи відображає метрики відповідності двох кривих. Інтерфейс зазначеної системи можна оцінити з рис. 1.3.

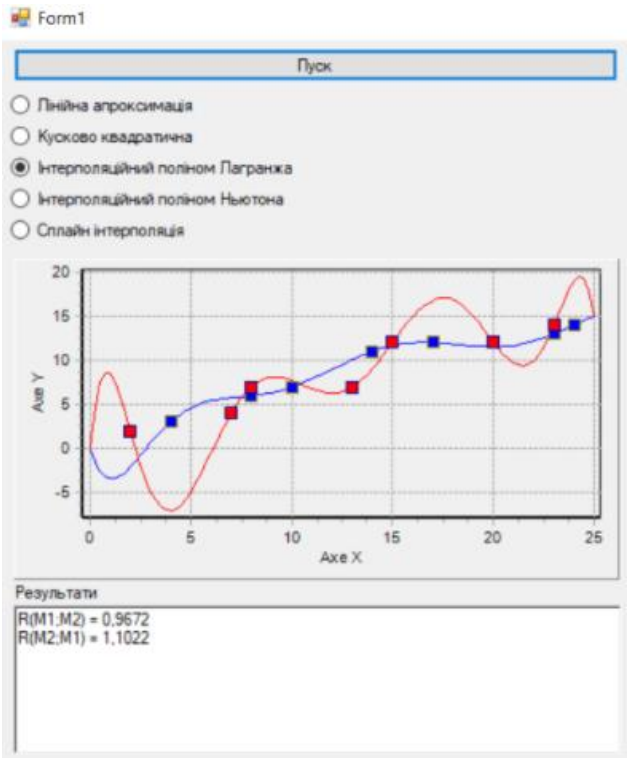


Рисунок 1.3 – Інтерфейс системи відповідності двох кривих

Основним недоліком «Системи оцінки подібності двох кривих» є її обмеженість щодо практичного застосування.

Ще однією автоматизованою системою, що має відношення до досліджуваної задачі, є онлайн система контролю руху кораблів «MarineTrafic».

«MarineTraffic» – це відкритий проект спільноти, який надає інформацію в режимі реального часу про переміщення суден і поточне розташування суден у портах. База даних інформації про судна включає, наприклад, деталі місць, де вони були побудовані, а також розміри суден, валовий тоннаж і номер Міжнародної морської організації (IMO). Користувачі можуть надсилати фотографії суден, які інші користувачі можуть оцінити [17].

Інтерфейс системи «MarineTraffic» можна оцінити з рис. 1.4.

Рисунок 1.4 – Інтерфейс системи «MarineTraffic»

До основних переваг даної системи належить:

- доступність (дана система є повністю безкоштовною);
- можливість відслідковувати рух кораблів у реальному часі;
- можливість отримання даних про будь-яке зареєстроване судно, його тип, назву та сполучення, за яким воно слідує.

До недоліків можна віднести те, що дана система не є експертною і не проводить жодного аналізу. Крім цього, дана інформаційна система має недостатню швидкодію та при її роботі часто виникають затримки і помилки. Основним недоліком системи є її недостатня захищеність, у зв’язку з чим вона

може піддаватись хакерським атакам, а дані отримані при одній з успішних атак можуть використовуватись як для продажу стороннім особам, так і для терористичних актів.

Автоматизованою системою, що має відношення до досліджуваної задачі, є і експертна система «FlightRadar24».

«FlightRadar24» − інтернет-проєкт, створений 2006 року шведською компанією Travel Network Ltd, який дозволяє в режимі реального часу спостерігати на карті рух літаків [18].

Дана система є аналогічною до системи «MarineTraffic» та має ті ж переваги і недоліки.

Інтерфейс системи «FlightRadar24» можна оцінити з рис. 1.5.

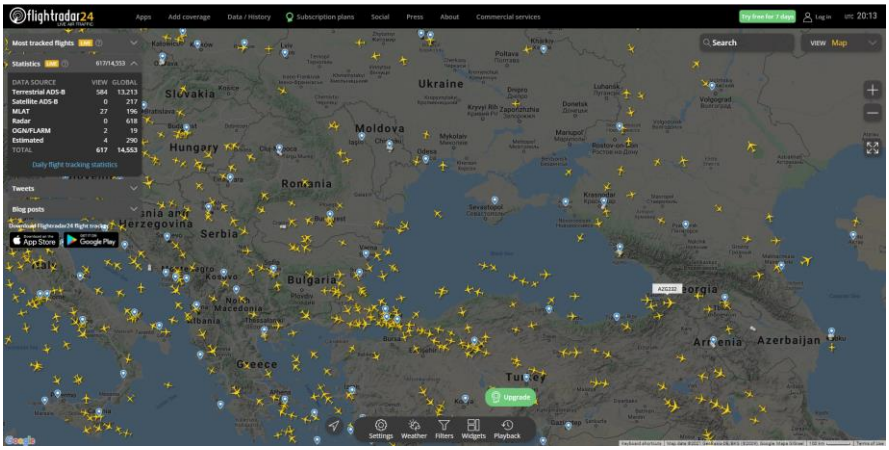


Рисунок 1.5 – Інтерфейс системи «FlightRadar24»

Проведений огляд програмних продуктів, що стосуються досліджуваної задачі, дозволяє зробити висновок про те, що наявні продукти мають суттєві недоліки щодо аспектів швидкодії, вартості, доступності, функціональної придатності. Це вказує на доцільність розробки для автоматизації вирішення досліджуваної у роботі задачі спеціалізованого програмного забезпечення.

1.3 Постановка задачі

Метою кваліфікаційної роботи бакалавра є створення методу для порівняння маршрутів кораблів відносно еталонного маршруту для систем висвітлення надводної обстановки та створення відповідної інформаційної системи для оцінки ступеня відхилення маршруту судна від еталонного маршруту та прийняття рішення щодо необхідності реагування на ймовірне порушення правил прикордонного законодавства. Застосунок може бути використаний суб'єктами забезпечення прикордонної безпеки для підвищення ефективності виявлення порушників прикордонного законодавства.

Розробку web-застосунку доцільно здійснити на платформі .NET Framework з використанням мови програмування C# та фреймворку даної мови – ASP .Net.

Відповідна інформаційна система призначена для автоматизації роботи таких користувачів, як оператор ЦУС регіонального управління Морської охорони, оператор Головного ЦУС Адміністрації Держприкордонслужби (див. рис. 1.1).

Для *оператора ЦУС регіонального управління Морської охорони* слід автоматизувати виконання наступних функцій:

- авторизацію оператора;
- роботу з реєстром акваторій, в межах яких проводяться спостереження;
- роботу з реєстром еталонних маршрутів суден;
- роботу з модулем оцінки маршрутів суден;
- роботу з реєстром спостережень;
- попередню оцінку загроз і ризиків щодо можливого порушення прикордонного законодавства;
- взаємодії з Головним ЦУС Адміністрації Держприкордонслужби.

Для *оператора Головного ЦУС Адміністрації Держприкордонслужби* слід автоматизувати виконання наступних функцій:

- авторизацію оператора;
- роботу з реєстром акваторій, в межах яких проводяться спостереження;
- роботу з реєстром еталонних маршрутів суден;

– роботу з модулем оцінки маршрутів суден;

– роботу з реєстром спостережень;

– оцінку загроз і ризиків щодо можливого порушення прикордонного законодавства;

– формування висновків за результатами оцінки загроз і ризиків;

– взаємодії з ЦУС регіонального управління Морської охорони.

2 Проектування структури інформаційної системи

Формування інформаційної системи для оцінки ступеня відхилення маршрутів кораблів від еталонного маршруту передбачає насамперед її проектування. Останнє потребує проведення аналізу досліджуваної задачі з фізичної точки зору, її формалізацію, виявлення визначальних характеристик процесів, усвідомлення питань щодо обробки інформаційних потоків, а також розробки структури інформаційної системи та вибір засобів для її створення.

2.1 Аналіз та автоматизація обробки інформаційних потоків

Діюча ІТС «Гарт-12» потребує удосконалення. Це пов’язано, зокрема, з недосконалістю методів, які застосовуються в ІТС для порівняння маршрутів двох суден. Остання обумовлена тим, що застосування лінійної апроксимації при формуванні неперервного апроксимаційного маршруту руху судна, є сумнівним і нічим не підтвердженим [14]. Особливий сумнів виникає у випадку значної віддаленості в часі або просторі досліджуваних точок, які характеризують положення судна. Тому актуальним завданням у межах проведення аналізу досліджуваної задачі з фізичної точки зору та її формалізації є дослідження особливостей встановлення величини метрики, що застосовується для ідентифікації подібності маршрутів суден у СВНО. При цьому, потребує особливої уваги завдання оцінки впливу виду апроксимації при побудові в СВНО неперервного маршруту руху судна через множину дискретних точок місцезнаходження судна у фіксовані моменти часу на величину метрики, що застосовується для встановлення подібності маршруту двох різних суден.

2.1.1. Аналіз та створення методу для вирішення досліджуваної задачі

У базі даних ІТС морської охорони «Гарт-12» інформація про пересування суден зберігається у вигляді впорядкованої послідовності точок

з координатами суден у певні моменти часу.

Уявлення про місце знаходження суден у фіксований момент часу можна сформувати, наприклад, з використанням рис. 2.1.

Сукупність координат точок місцезнаходження *i*-го судна можна позначити впорядкованою множиною (кортежем)

$$M_i = \{(x_1, y_1),\ldots, (x_{\mathrm{ki}}, y_{\mathrm{ki}})\}, \quad (2.1)$$

де k_{i} – потужність множини M_i.

Слід зауважити, що потужності відповідних множин для різних суден, у загальному випадку, різні. Причому потужності цих множин можуть відрізнятися достатньо суттєво навіть у випадку, якщо судна рухаються по одному маршруту між заданими пунктами відправлення та призначення.

Траєкторії руху суден у вигляді неперервних кривих відсутні.

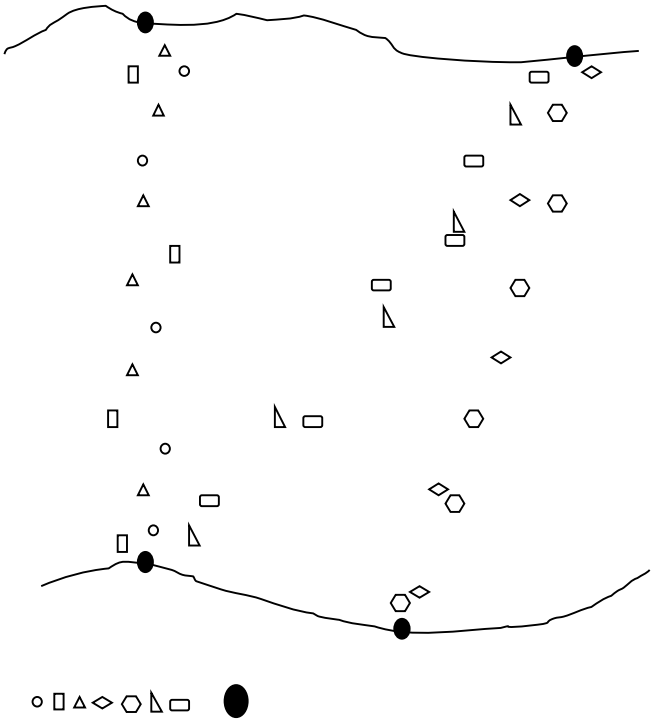


Рисунок 2.1 – Схематичне представлення даних щодо положення окремих суден у СВНО:

Дослідження питання про можливість порушення судном правил

прикордонного режиму у загальному випадку передбачає вирішення комплексу взаємопов’язаних задач:

1. встановлення подібності маршрутів довільних пар різних суден;

2. встановлення кластеру маршрутів між заданими пунктами відправлення та призначення (транспортного коридору між двома пунктами);

3. визначення еталонного маршруту між двома пунктами в межах встановленого кластеру;

4. встановлення належності траєкторії руху судна кластеру маршрутів між заданими пунктами відправлення та призначення;

5. оцінка ступеню відхилення траєкторії досліджуваного судна від еталонного маршруту.

У межах даної роботи аналізу піддаватимуться лише перші дві задачі.

Задача 1.

Нехай задано маршрути руху деякої пари суден у вигляді множин M_1, M_2 типу (2.1).

Ступінь подібності маршрутів M_1 та M_2 (див. рис. 2.2) у роботі [19] запропоновано визначати за допомогою метрики $R(M_1, M_2)$.

Для розрахунку $R(M_1, M_2)$ необхідно насамперед здійснити встановлення (відновлення, генерування) неперервного маршруту руху першого судна між заданими у дискретній множині M_1 точками з використанням лінійної апроксимації (див. рис 2.2). Величина $R(M_1, M_2)$ визначається, як середнє відхилення точок множини M_2, які описують дискретний маршрут руху другого судна, від апроксимованого неперервного першого маршруту у вигляді

$$R(M_1, M_2) = \frac{\sum_{j=1}^{k_2} R_{\min j}}{k_2}, \qquad (2.2)$$

де $R_{\min j}$ є найкоротшою відстанню від j-ї точки множини M_2 до встановленого першого маршруту, тобто до найближчого деякого i-го сегменту прямої, який з’єднує точки (x_i, y_i) і (x_{i+1}, y_{i+1}) множини M_1, а k_2 - кількість досліджуваних точок другого маршруту.

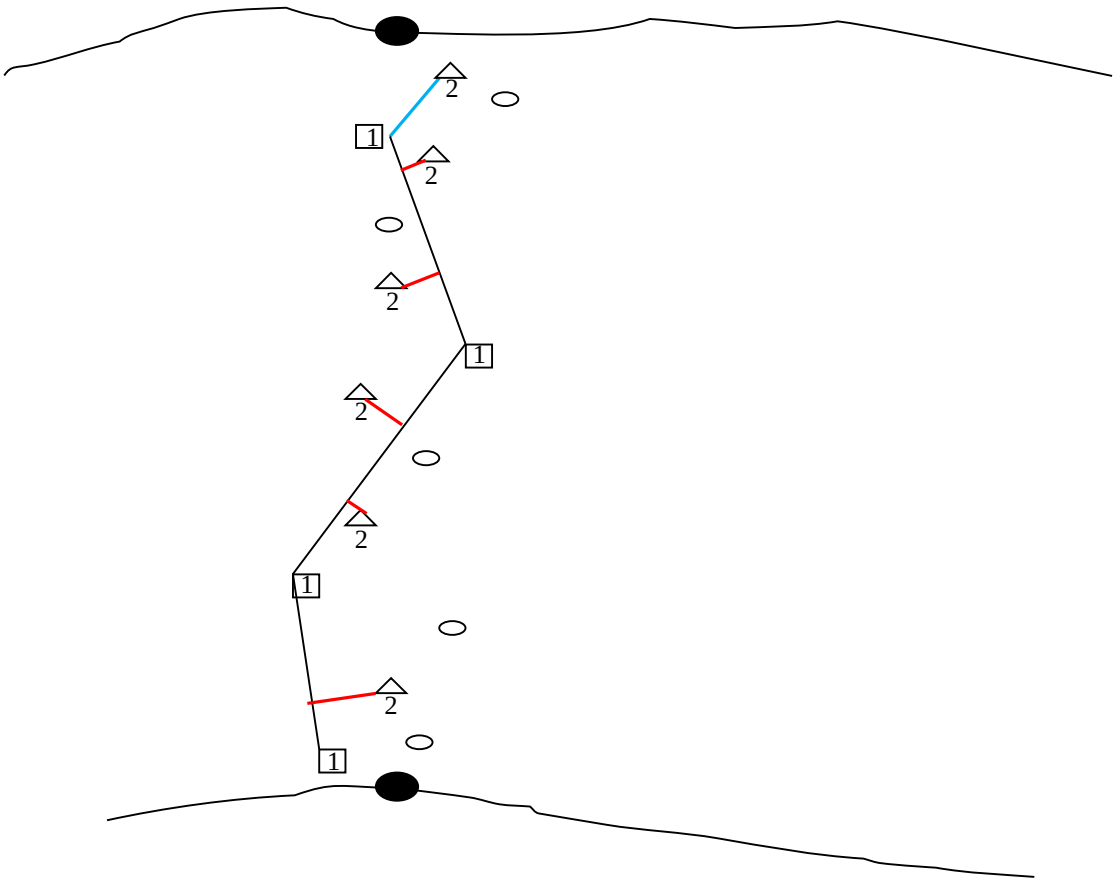


Рисунок 2.2 – Підхід до визначення ступеня подібності маршрутів за допомогою метрики

При використанні лінійної апроксимації сегментів першого маршруту координати (x,y) точки i-го сегменту прямої, який з'єднує точки (x_i,y_i) і (x_{i+1},y_{i+1}) першого маршруту, яка є найближчою до j-ї точки множини M_2, координати якої рівні (x_j,y_j), можна отримати з формул

$$x = \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1 + k_i^2}, \tag{2.3}$$

$$y = k_i \cdot x + b_i,$$

де k_i, b_i – параметри лінійного рівняння прямої, що описує i-й сегмент, який з'єднує точки (x_i,y_i) і (x_{i+1},y_{i+1}) множини M_1. Вказані параметри можуть бути нескладно знайдені з використанням рівняння прямої, що проходить через дві задані точки.

Тоді

$$R_{\min j} = \sqrt{(x_j - \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1+k_i^2})^2 + (y_j - k_i \cdot \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1+k_i^2} - b_i)^2}. \quad (2.4)$$

Формула (2.4) може застосовуватись у випадку, коли найкоротша відстань від j-ї точки множини M_2 до встановленого першого маршруту знаходиться на перпендикулярі від j-ї точки множини M_2 до деякого сегмента неперервного апроксимованого першого маршруту (на рис. 2.2 це відрізки, що зображені червоним кольором).

Однак, можливий випадок, коли величина x, що отримана з (2.3), може знаходитись поза діапазоном $[x_i, x_{i+1}]$. Тоді величину $R_{\min j}$ пропонується знаходити як найменшу відстань від j-ї точки множини M_2 до кінців відрізку i-го сегменту (тобто до точок (x_i, y_i) і (x_{i+1}, y_{i+1}) множини M_1) (на рис. 2.2 це відрізок, що зображений синім кольором).

Тоді, з урахуванням цього, у загальному випадку

$$R_{\min j} = \begin{cases} \sqrt{(x_j - \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1+k_i^2})^2 + (y_j - k_i \cdot \frac{y_j \cdot k_i + x_j - b_i \cdot k_i}{1+k_i^2} - b_i)^2}, x \in [x_i, x_{i+1}], \\ \min\left\{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \sqrt{(x_j - x_{i+1})^2 + (y_j - y_{i+1})^2}\right\}, x \notin [x_i, x_{i+1}]. \end{cases} . \quad (2.5)$$

Послідовне використання виразів (2.5), (2.3), (2.2) визначає методику обчислення метрики $R(M_1, M_2)$.

Слід зауважити, що величина метрики R, яка використовується для оцінки ступеня подібності першого та другого маршрутів залежить від порядку слідування аргументів у записі для R. Так, якщо в записі $R(M_1, M_2)$ аргументи M_1 та M_2 поміняти місцями, то запис метрики матиме вигляд $R(M_2, M_1)$, а її відшукання для того ж випадку початкових даних, що й наведені на рис. 2.2, може бути оцінене з рис. 2.3.

Для розрахунку $R(M_2, M_1)$ необхідно насамперед здійснити встановлення (відновлення, генерування) неперервного маршруту руху другого судна між заданими у дискретній множині M_2 точками з використанням лінійної апроксимації (див. рис. 2.3). Тоді величина $R(M_2, M_1)$ визначається, як середнє відхилення точок множини M_1, які описують дискретний маршрут руху першого судна, від апроксимованого неперервного

другого маршруту у вигляді

$$R(M_2, M_1) = \frac{\sum_{j=1}^{k_1} R_{\min j}}{k_1}, \qquad (2.6)$$

де $R_{\min j}$ є найкоротшою відстанню від j-ї точки множини M_1 до встановленого другого маршруту, тобто до найближчого деякого i-го сегменту прямої, який з'єднує точки (x_i,y_i) і (x_{i+1},y_{i+1}) множини M_2, а k_1 - кількість досліджуваних точок першого маршруту.

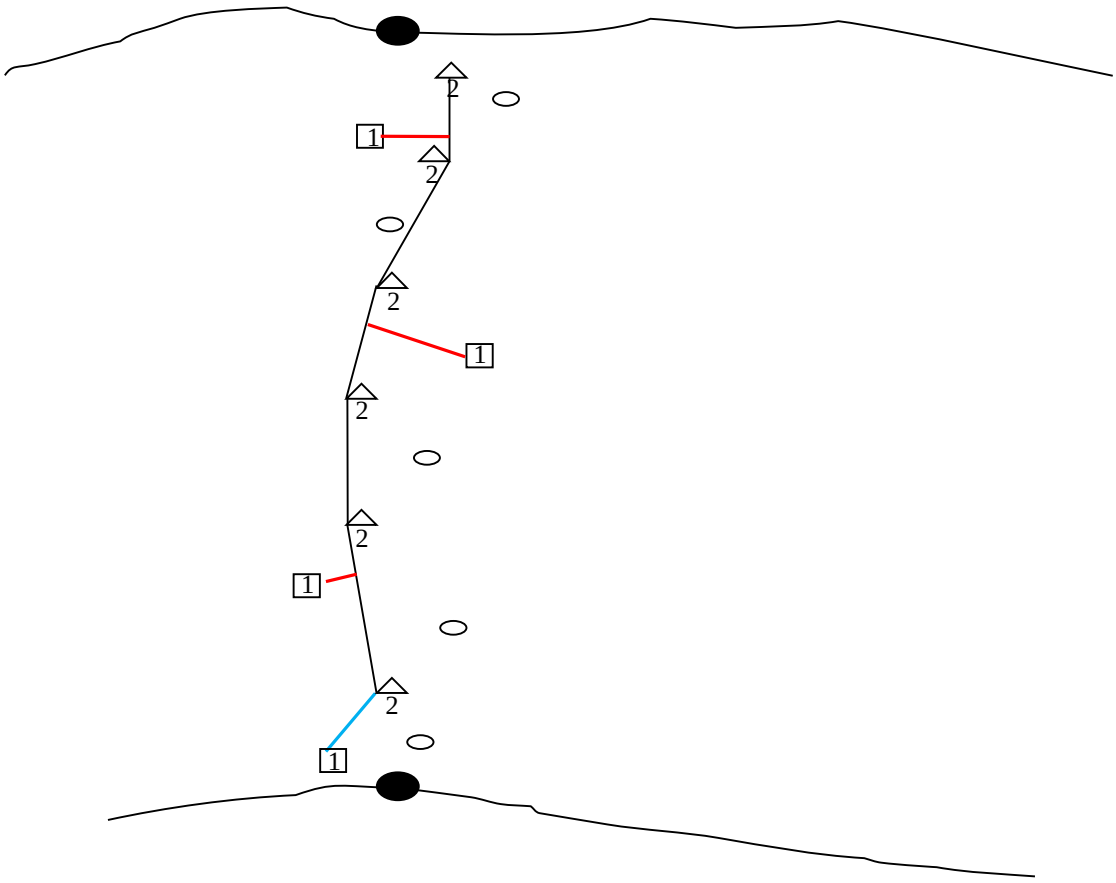


Рисунок 2.3 – Підхід до визначення ступеня подібності маршрутів за допомогою метрики

При використанні лінійної апроксимації сегментів другого маршруту координати (x,y) точки i-го сегменту прямої, який з'єднує точки (x_i,y_i) і (x_{i+1},y_{i+1}) другого маршруту, яка є найближчою до j-ї точки множини M_1, координати якої рівні (x_j,y_j), також можна отримати за допомогою формул (2.3), але з урахуванням того, що k_i, b_i – параметри лінійного рівняння прямої,

що описує i-й сегмент, який з'єднує точки (x_i,y_i) і (x_{i+1},y_{i+1}) множини M_2. Вказані параметри можуть бути нескладно знайдені з використанням рівняння прямої, що проходить через дві задані точки.

Тоді $R_{\min j}$ можна визначати за допомогою формули (2.4) у випадку, коли найкоротша відстань від j-ї точки множини M_1 до встановленого другого маршруту знаходиться на перпендикулярі від j-ї точки множини M_1 до деякого сегмента неперервного апроксимованого другого маршруту (на рис. 2.3 це відрізки, що зображені червоним кольором).

У випадку ж, коли величина x, що отримана з (2.3), знаходиться поза діапазоном $[x_i,\ x_{i+1}]$, величину $R_{\min j}$ пропонується знаходити як найменшу відстань від j-ї точки множини M_1 до кінців відрізку i-го сегменту (тобто до точок (x_i,y_i) і (x_{i+1},y_{i+1}) множини M_2) (на рис. 2.3 це відрізок, що зображений синім кольором).

Тоді, з урахуванням цього, у загальному випадку $R_{\min j}$ буде також знаходитись за допомогою формули (2.5).

Аналіз підходів і даних, що наведені на рис. 2.2, 2.3, спонукає до необхідності перевірки рівності метрик $R(M_1,M_2), R(M_2,M_1)$. Адже, для однієї і тієї ж пари маршрутів у залежності від обраної метрики у разі неспівпадання $R(M_1,M_2)$ і $R(M_2,M_1)$ можуть мати місце різні висновки щодо подібності чи неподібності маршрутів. Саме тому для перевірки рівності величин $R(M_1,M_2)$ і $R(M_2,M_1)$ у роботі [19] було проведено окреме дослідження для вихідних даних, які наведені на рис. 2.4.

Застосування формул (2.2)-(2.6) для початкових даних рис. 2.4 дозволило отримати, що $R(M_1,M_2)$=18, $R(M_2,M_1)$=26. А отже, у загальному випадку, дійсно $R(M_1,M_2) \neq R(M_2,M_1)$.

Відсутність властивості комутативності щодо метрик $R(M_1,M_2)$ і $R(M_2,M_1)$ спонукала авторів роботи [19] до формування пропозицій щодо застосування нової метрики, яка б не містила недоліку, що стосувався б неоднозначності формування висновків відносно подібності маршрутів.

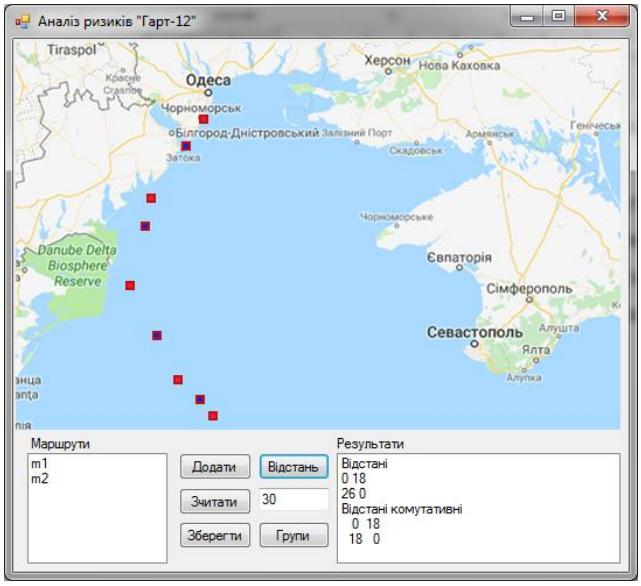
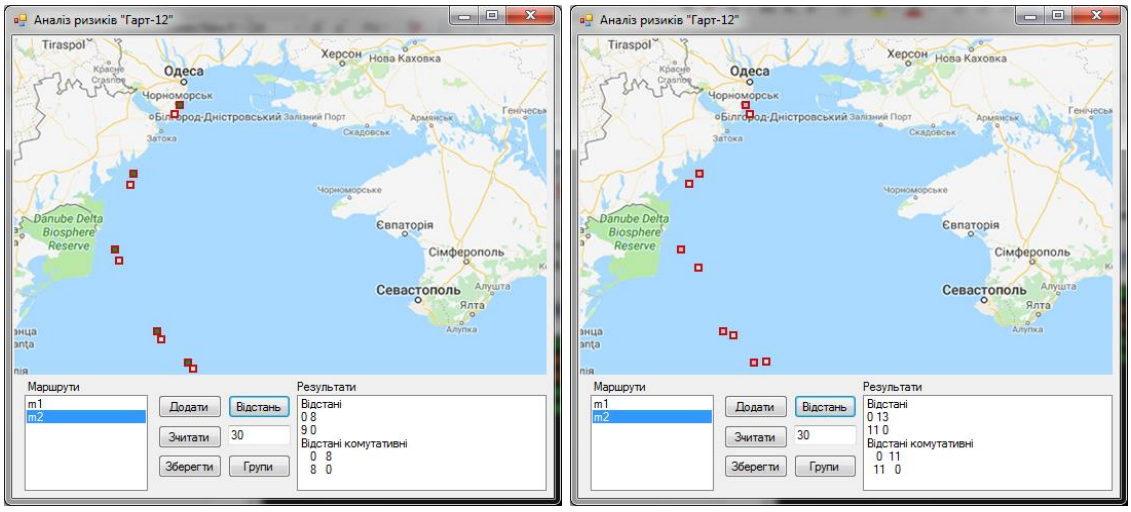


Рисунок 2.4 – Розрахунок метрик $R(M_1,M_2)$ і $R(M_2,M_1)$

Ця пропозиція має вигляд метрики виду

$$R^k(M_1, M_2) = \min(R(M_1, M_2), R(M_2, M_1)). \quad (2.7)$$

Для перевірки властивості комутативності для метрики (2.7) було проведено дослідження для початкових даних, що наведені на рис. 2.5. Результати дослідження, що описані у праці [21], вказують на комутативність метрики (2.7).



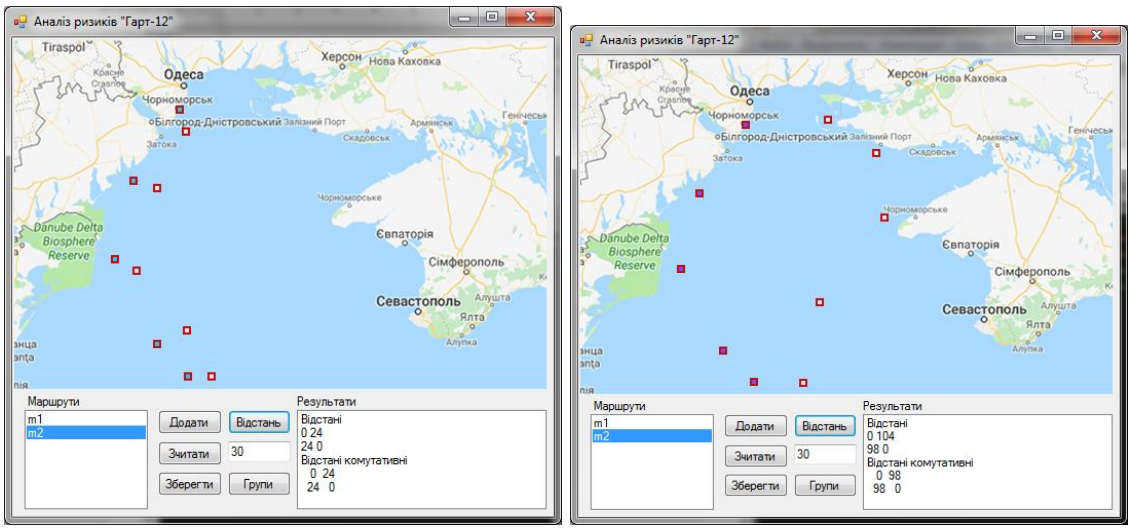


Рисунок 2.5 – Приклади обчислення метрики (2.7)

Однак питанню об’єктивності результатів, отриманих за допомогою метрики (2.7), у роботі [21] увага не приділялася.

Задача 2.

Введення деякого граничного порогового рівня r_{pk} (даний рівень визначається на основі експертної оцінки) щодо величини відхилення маршрутів і його порівняння з метрикою (2.7) дозволяє робити висновки про ступінь подібності маршрутів M_1 і M_2 незалежно від того, для якого з маршрутів формується апроксимаційна неперервна траєкторія руху, і відносно якого здійснюється оцінка близькості до такої траєкторії.

Встановлення геометричної близькості маршрутів за допомогою формули (2.7) дає можливість провести їх подальшу класифікацію та поділ на кластери (кластеризацію) з подальшим визначенням у кожному з них еталонного маршруту. Еталонні маршрути можуть використовуватись для оцінки ступеня відхилення судна, відносно якого проводиться аналіз. Також може здійснюватись подальша класифікація нових даних, що поступають до системи, та оновлення або уточнення меж кластера.

Для проведення кластеризації маршрутів на основі наявних у базі «Гарт-12» даних у роботі [21] пропонується побудувати матрицю $\boldsymbol{R}$, елементи r_{ij} якої визначаються наступним чином

$$r_{\text{ij}} = \begin{cases} 0, i = j, \\ R^k(M_i, M_j), i \neq j, \end{cases}. \quad (2.8)$$

де M_i, M_j - маршрути відповідно i-го та j-го суден.

Після формування матриці $\boldsymbol{R}$ для кожного її рядка, тобто для $i = \text{const}$, обчислюється:

- кількість n_{s} елементів, які не перевищують порогового для кластеризації рівня r_{pk}, тобто кількість елементів, для яких у рядку i виконується нерівність $R^k(M_i, M_j) \leq r_{\text{pk}}$;

- величина $s_v = \sum R^k(M_i, M_j)$, яка формується лише з тих значень $R^k(M_i, M_j)$, для яких виконується умова $R^k(M_i, M_j) \leq r_{\text{pk}}$.

На основі отриманих даних, які сортуються за зменшенням n_{s} (для елементів з однаковими значеннями n_{s} здійснюється сортування за збільшенням s_{v}), формується впорядкована множина $\boldsymbol{M}_{\text{s}}$, елементами якої є набори (i, $n_{\text{s}}(i)$, $s_{\text{v}}(i)$).

Далі множина $\boldsymbol{M}_{\text{s}}$ використовується для визначення тих маршрутів, які можуть бути включені до одного і того ж кластера. Здійснюється це за допомогою наступного алгоритму:

1. Якщо для першого елемента множини $\boldsymbol{M}_{\text{s}}$ виконується умова $n_{\text{s}}(i)>0$, то створюється кластер з центральним маршрутом i і в нього включаються всі маршрути j, для яких $R^k(M_i, M_j) \leq r_{\text{pk}}$.

2. З множини $\boldsymbol{M}_{\text{s}}$ видаляються всі набори (i, $n_{\text{s}}(i)$, $s_{\text{v}}(i)$), в яких перший елемент i співпадає із значеннями j, які визначили маршрути, що увійшли до створеного кластера.

3. Кроки 1-2 алгоритму повторюються до тих пір, поки потужність множини $\boldsymbol{M}_{\text{s}}$ є більшою за 0.

Реалізація такого підходу дозволяє встановити всі кластери (коридори руху суден), які задовольняють умові не перевищення порогу r_{pk}.

Проведений вище аналіз задач 1, 2 дозволяє зробити висновки про те, що у них є багато моментів, які можуть впливати на достовірність результатів

моделювання та зменшувати ефективність автоматизованої системи обробки даних СВНО.

До числа таких можна віднести:

1. Припущення про доцільність застосування лінійної апроксимації при роботі з окремими сегментами між точками множини $M_i = \{(x_1, y_1),..., (x_{\text{ki}}, y_{\text{ki}})\}$ при формуванні неперервного апроксимаційного маршруту чи M_1, чи M_2, є сумнівним і нічим не підтвердженим. Особливий сумнів виникає у випадку значної віддаленості в часі або просторі досліджуваних точок. А тому актуальності набуває завдання дослідження достовірності результатів на заданих експериментальних наборах даних у разі, якщо застосовуватиметься інший вид апроксимації (в тому числі за допомогою сплайнів). При цьому, окремої уваги потребує питання відповідності обраного методу апроксимації логічному змісту процесу руху суден.

2. Дослідження маршрутів різних суден на предмет формування кластерів може здійснюватись не лише в межах між пунктами відправлення та призначення, а й між довільними двома точками, в межах яких існує особлива необхідність вивчення питання про порушення прикордонного законодавства. У цьому випадку застосування формули (2.2) для відшукання метрики $R(M_1, M_2)$, формули (2.6) для відшукання метрики $R(M_2, M_1)$, або формули (2.7) для встановлення метрики $R^k(M_1, M_2)$, потребує обгрунтування вибору кількостей k_2 і k_1 необхідних для дослідження точок, які б забезпечили достатній рівень достовірності висновків, що формуються.

3. Для умов, що наведені вище, актуальним є питання і вибору для дослідження конкретних точок на маршрутах M_1 та M_2, які при реалізації значень k_2 і k_1 забезпечували б достатній рівень достовірності висновків, що формуються. Актуальність дослідження цього питання обумовлюється тим, що при реалізації різних комбінацій досліджуваних точок величина $R(M_1, M_2)$, або $R(M_2, M_1)$, або $R^k(M_1, M_2)$ може суттєво відрізнятися, що може впливати на достовірність методики кластеризації.

4. Викликає сумнів і питання забезпечення об’єктивності результатів,

що визначаються застосуванням метрики (2.7). Адже властивість комутативності метрики справедлива не лише у випадку застосування (2.7). Так, при застосуванні метрики

$$R^k(M_1, M_2) = \max(R(M_1, M_2), R(M_2, M_1)), \quad (2.9)$$

або

$$R^k(M_1, M_2) = \frac{R(M_1, M_2) + R(M_2, M_1)}{2}, \quad (2.10)$$

або

$$R^k(M_1, M_2) = \sqrt{R(M_1, M_2) R(M_2, M_1)}, \quad (2.11)$$

існує велика ймовірність її комутативності, а отже, можливості застосування. Однак висновки щодо подібності маршрутів та співпадання меж кластерів при застосуванні наведених метрик у різних випадках можуть відрізнятися. А отже, питання адекватності застосовуваного інструментарію та достовірності результатів, що ним забезпечуються, потребують окремого вивчення.

5. При застосуванні метрик (2.2), або (2.6), або (2.7) може мати місце випадок, коли для окремих сегментів між точками (x_i, y_i) і (x_{i+1}, y_{i+1}) множини M_1 або M_2 виконуватиметься умова $x_i = x_{i+1}$. Тобто у вибраній системі координат окремі сегменти можуть бути паралельними осі ординат. У цьому випадку формула (2.5) не може бути застосовною для відшукання величини $R_{\min j}$.

6. У разі, якщо деякий досліджуваний маршрут M_i на окремій ділянці маршруту між пунктом відправлення та призначення паралельний еталонному маршруту M_{et}, але на цій ділянці величина $R(M_i, M_{\text{et}}) > r_{\text{pk}}$, то це може спричинити не включення маршруту M_i у кластер, хоча у загальному випадку цей маршрут може увійти до цього кластеру. Разом з тим, існуюча методика кластеризації цієї особливості не враховує. А оскільки така особливість важлива з позиції формування висновків щодо наявності порушень прикордонного законодавства, то відповідне питання також потребує окремого дослідження.

7. У разі достатньої складності маршруту існуюча методика може бути

незастосовною, оскільки може не відповідати логічному змісту досліджуваного процесу.

8. Формування еталонного маршруту M_{et} описане у роботі [23]. Цей маршрут являє собою набір дискретних точок. Актуальним є питання формування такого методу побудови еталонного маршруту M_{et}, який би забезпечував створення еталонного маршруту у вигляді неперервної траєкторії.

2.1.2. Аналіз питань щодо обробки інформаційних потоків

Автоматизована інформаційна система для оцінки ступеня відхилення маршруту судна від еталонного маршруту призначена для автоматизації роботи двох видів користувачів – оператора ЦУС регіонального управління Морської охорони та оператора Головного ЦУС Адміністрації Держприкордонслужби.

Згідно умови досліджуваної задачі для *оператора ЦУС регіонального управління Морської охорони* слід автоматизувати виконання наступних функцій:

- авторизацію оператора;
- роботу з реєстром акваторій, в межах яких проводяться спостереження;
- роботу з реєстром еталонних маршрутів суден;
- роботу з модулем оцінки маршрутів суден;
- роботу з реєстром спостережень;
- попередню оцінку загроз і ризиків щодо можливого порушення прикордонного законодавства;
- взаємодії з Головним ЦУС Адміністрації Держприкордонслужби.

Для *оператора Головного ЦУС Адміністрації Держприкордонслужби* слід автоматизувати виконання наступних функцій:

- авторизацію оператора;
- роботу з реєстром акваторій, в межах яких проводяться спостереження;
- роботу з реєстром еталонних маршрутів суден;
- роботу з модулем оцінки маршрутів суден;

– роботу з реєстром спостережень;

– оцінку загроз і ризиків щодо можливого порушення прикордонного законодавства;

– формування висновків за результатами оцінки загроз і ризиків;

– взаємодії з ЦУС регіонального управління Морської охорони.

Функція «Авторизація оператора», «Робота з реєстром акваторій, в межах яких проводяться спостереження», «Робота з реєстром еталонних маршрутів суден», «Робота з модулем оцінки маршрутів суден», «Робота з реєстром спостережень» є спільними для обох видів користувачів.

Функція «Авторизація оператора» призначений для ідентифікації користувача в системі. Виконується шляхом реалізації наступних кроків:

1. користувач обирає відповідний пункт меню «Авторизація оператора»;

2. користувач заповнює форму;

3. виконується запит до бази даних, до таблиці, у якій зберігається інформація про користувачів;

4. відбувається визначення виду користувача;

5. при успішній авторизації відбувається відображення кабінету оператора.

Функція «Робота з реєстром акваторій, в межах яких проводяться спостереження» призначений для перегляду, додавання, редагування та видалення акваторій. Виконується шляхом реалізації наступних кроків:

1. авторизований оператор вибирає відповідний пункт меню «Робота з реєстром акваторій, в межах яких проводяться спостереження»;

2. виконується запит до бази даних для перегляду акваторій з наявного переліку. Користувачеві відображається веб-сторінка з їх переліком;

3. користувач обирає одну з дій, яку бажає виконати (перегляд, додавання, редагування, видалення акваторії);

4. при виборі дії «перегляд» відбувається запит до бази даних для отримання даних про акваторію. При виборі дії «додавання» чи «редагування» користувачу необхідно заповнити форму, далі виконується відповідний запит до

бази даних та відображення відповідної веб-сторінки. При виборі дії «видалення» відбувається запит до бази даних і відповідна акваторія видаляється.

Функція «Робота з реєстром еталонних маршрутів суден» призначений для автоматизації перегляду переліку еталонних маршрутів користувачем. Виконується шляхом реалізації наступних кроків:

1. авторизований користувач вибирає відповідний пункт меню «Робота з реєстром еталонних маршрутів суден»;

2. виконується запит до бази даних для вибору маршруту з переліку існуючих для раніше обраної акваторії. Відображається відповідна веб-сторінка з переліком еталонних маршрутів та датою коли вони були створені;

3. користувач обирає еталонний маршрут, щодо якого він хотів би отримати повну інформацію;

4. виконується запит до бази даних про конкретний маршрут. Відображається веб-сторінка із даними отриманими у запиті;

5. відбувається побудова графіку, що відображає даний еталонний маршрут.

Функція «Робота з модулем оцінки маршрутів суден» призначений для автоматизації відображення маршрутів суден. Виконується шляхом реалізації наступних кроків:

1. вибір сукупності дискретних точок, що визначають місцезнаходження судна в окремі моменти часу;

2. побудова за обраною сукупністю точок траєкторії руху судна.

Функція «Робота з реєстром спостережень» призначений для автоматизації процесу перегляду користувачем реєстру спостережень, що проводилися раніше. Даний бізнес-процес реалізується таким чином:

1. авторизований користувач вибирає відповідний пункт меню «Робота з реєстром спостережень»;

2. виконується запит до бази даних для вибору тих маршрутів суден, що відповідають відповідним пунктам відправки та призначення;

3. відображається відповідна веб-сторінка із переліком спостережень, їх коротким описом та датою коли вони були проведені;

4. користувач обирає маршрут, щодо якого він хотів би отримати повну інформацію;

5. виконується запит до бази даних для вибору даних про конкретне спостереження;

6. відображається веб-сторінка із даними отриманими у запиті.

Функція «Попередня оцінка загроз і ризиків щодо можливого порушення прикордонного законодавства» призначений для автоматизації порівняння маршрутів суден з еталонними маршрутами. Виконується шляхом реалізації наступних кроків:

1. користувачем обирається набір еталонних маршрутів між тими ж пунктами відправки та призначення і які на його думку подібні за типом до досліджуваного судна;

2. реалізується процедура побудови кластера подібних маршрутів та нового еталонного маршруту;

3. реалізується процедура візуалізації еталонного маршруту, кластера подібних маршрутів, траєкторії досліджуваного судна;

4. реалізується процедура порівняння маршруту судна з еталонним маршрутом на основі обробки даних згідно теорії, що наведена в п. 2.1.1;

5. на основі порівняння об’єктів пп. 3, 4 здійснюється вибір одного з попередніх висновків «траєкторія судна не викликає підозр», «траєкторія судна викликає підозри».

Функція «Взаємодія з Головним ЦУС Адміністрації Держприкордонслужби» призначений для автоматизації передачі даних до інстанції, яка проводить детальний аналіз підозрілих ситуацій і формує рішення щодо адекватного реагування. Виконується шляхом реалізації наступних кроків:

1. здійснюється надсилання до Головного ЦУС Адміністрації Держприкордонслужби даних з ідентифікаційною маскою «траєкторія судна викликає підозри»;

2. вноситься запит на прийняте рішення;

3. опрацьовується отримана на запит відповідь;

4. здійснюється в разі необхідності передача вказівок щодо реагування відповідним суб’єктам;

5. здійснюється надсилання до Головного ЦУС Адміністрації Держприкордонслужби інформації про реалізацію рішень.

Функція «Оцінка загроз і ризиків щодо можливого порушення прикордонного законодавства» призначений для автоматизації порівняння маршрутів кораблів з еталонними маршрутами. Виконується шляхом реалізації наступних кроків:

1. користувачем обирається набір маршрутів тих суден, які рухалися між тими ж пунктами відправки та призначення і які на його думку подібні за типом до досліджуваного судна;

2. реалізується процедура побудови кластера подібних маршрутів та нового еталонного маршруту;

3. реалізується процедура візуалізації еталонного маршруту, кластера подібних маршрутів, траєкторії досліджуваного судна;

4. реалізується процедура порівняння маршруту судна з еталонним маршрутом на основі обробки даних згідно теорії, що наведена в п. 2.1.1;

5. на основі порівняння об’єктів пп. 3, 4 здійснюється вибір одного з попередніх висновків «траєкторія судна не викликає підозр», «траєкторія судна викликає підозри».

6. завантажуються результати реалізації бізнес-процесу «Попередня оцінка загроз і ризиків щодо можливого порушення прикордонного законодавства».

Функція «Формування висновків за результатами оцінки загроз і ризиків» призначений для автоматизації формування висновку щодо необхідності реагування на можливе порушення судном прикордонного законодавства. Виконується шляхом реалізації наступних кроків:

1. здійснюється порівняння висновків, що отримані за результатами реалізації бізнес-процесів «Попередня оцінка загроз і ризиків щодо можливого порушення прикордонного законодавства» і «Оцінка загроз і ризиків щодо можливого порушення прикордонного законодавства»;

2. на основі обробки висновків, отриманих у результаті реалізації п. 1, формується один з висновків «реагування недоцільне», або «необхідне реагування».

Функція «Взаємодія з ЦУС регіонального управління Морської охорони» призначений для автоматизації процедур щодо реагування на наявні факти можливого порушення прикордонного законодавства. Виконується шляхом реалізації наступних кроків:

1. здійснюється надсилання до ЦУС регіонального управління Морської охорони даних з ідентифікаційною маскою «траєкторія судна задовільна», або «траєкторія судна потребує реагування»;

2. здійснюється запит на прийняті рішення щодо застосування суб’єктів реагування.

На рис. 2.6 зображено діаграму активності оператора ЦУС регіонального управління Морської охорони при роботі із модулем дослідження відповідності маршруту судна еталонному маршруту.

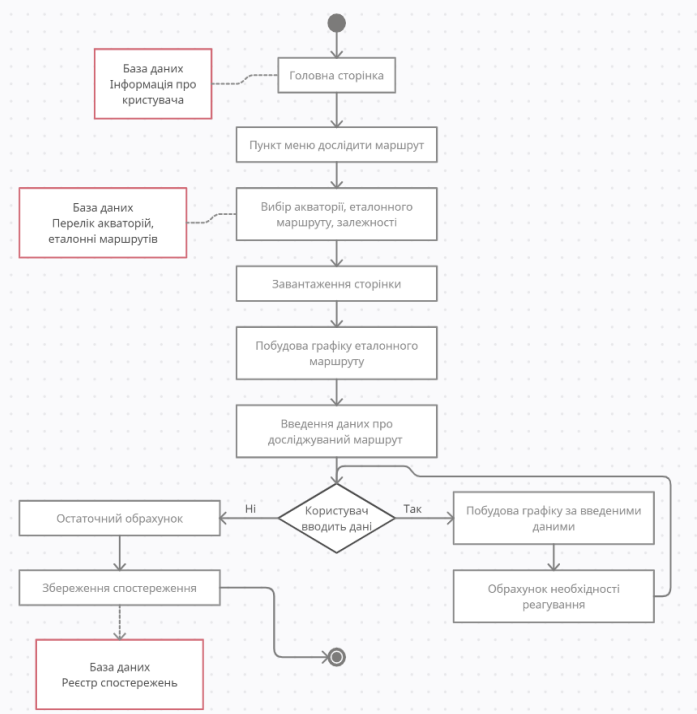


Рисунок 2.6 – Діаграма активності при дослідженні маршруту

2.2 Розробка структури інформаційної системи

Для розробки структури бази даних автоматизованої інформаційної системи оцінки ступеня відхилення маршруту судна від еталонного маршруту необхідно визначити сутності предметної області та їх властивості. Створення бази даних завжди починається з розробки структури її таблиць. Структура повинна бути такою, щоб при роботі з базою даних необхідно було б вводити в неї якомога менше даних.

Для назв таблиць сутностей і їх властивостей використовувалась латиниця, щоб полегшити роботу із SQL запитами. На рис. 2.7 зображена даталогічна модель бази даних досліджуваної АІС, яка передбачає такі таблиці: англ: Water_areas (акваторії), Observations (спостереження), Water_areas_types (типи акваторій), Best_routes (еталонні маршрути), User_types (типи користувачів інформаційної системи), Users (оператори інформаційної системи), Best-routes-ships_pivot (розвідна таблиця між еталонними маршрутами та типами

суден, для яких даний маршрут призначений), Ship_types (типи суден), Dependencies (математичні залежності), Messages (повідомлення).

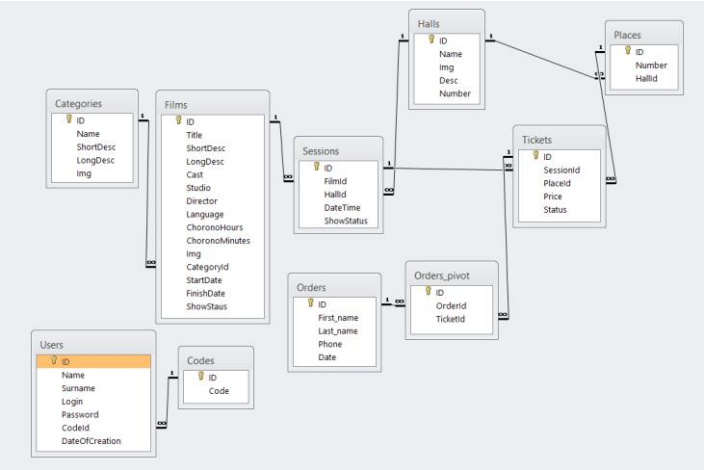


Рисунок 2.7 – Даталогічна структура бази даних АІС оцінки ступеня відхилення маршруту судна від еталонного маршруту

Також окрім розробки структури бази даних було розроблено схему роботи користувача із автоматизованою інформаційною системою (рис 2.8).

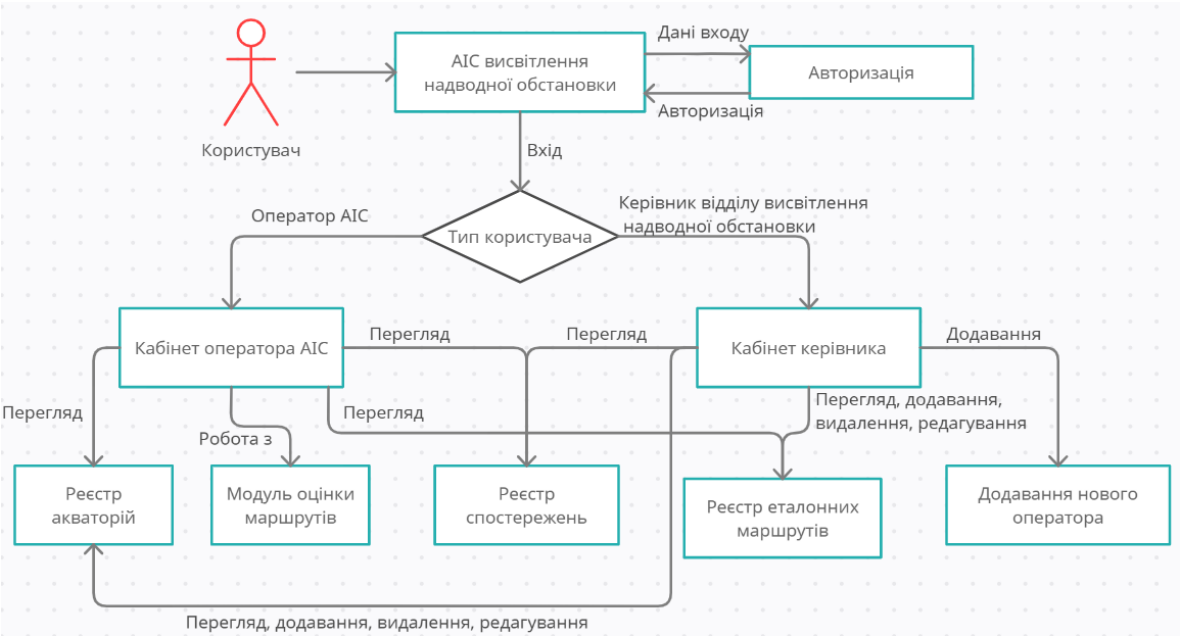


Рисунок 2.8 – Схема роботи користувача з інформаційною системою

Структура кожної таблиці бази даних розроблялась окремо з урахуванням особливостей, що стосуються тієї, чи іншої сутності. Тому було створено структуру бази даних, необхідну для функціонування автоматизованої інформаційної системи.

Таблиця «Water_areas» призначена для зберігання інформації про акваторії, їх координати та опис даної акваторії.

Таблиця 2.1 – Атрибути таблиці «Water_areas»

№ з/п	Назва атрибуту	Тип даних	Опис
1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Name	VARCHAR	Назва акваторії
3.	Countries	VARCHAR	Назви країн, які мають доступ до даної акваторії
4.	Coordinates	VARCHAR	Координати акваторії
5.	Description	VARCHAR	Опис акваторії
6.	Type_id	INT	Ідентифікатор типу акваторії
7.	Created_at	DATE	Дата, коли було створено замовлення

Таблиця «Water_areas_types» призначена для зберігання типів акваторій, тому основними атрибутами даної сутності є її ідентифікатор на назва типу акваторії.

Таблиця 2.2 – Атрибути таблиці «Water_areas_types»

№ з/п	Назва атрибуту	Тип даних	Опис
1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Type	VARCHAR	Назва типу акваторії

Таблиця «Users» призначена для користувачів, які зареєстровані в АІС та можуть починати роботу із інформаційною системою після авторизації.

Таблиця 2.3 – Атрибути таблиці «Users»

№ з/п	Назва атрибуту	Тип даних	Опис
1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Name	VARCHAR	Ім’я зареєстрованого користувача
3.	Surname	VARCHAR	Прізвище зареєстрованого користувача
4.	Login	VARCHAR	Логін, що використовується для входу
5.	Password	VARCHAR	Пароль, що використовується для входу
6.	Type_id	INT	Тип користувача
7.	Created_at	DATETIME	Дата створення акаунту

Таблиця «User_types» призначена для зберігання типів користувачів, саме згідно даної таблиці відбувається розподілення користувачів за типами та відображення їх відповідно різних електронних кабінетів.

Таблиця 2.5 – Атрибути таблиці «User_types»

№ з/п	Назва атрибуту	Тип даних	Опис
1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Type	VARCHAR	Назва типу користувача

Таблиця «Ship_types» призначена для зберігання назв типів суден, використовуючи інформацію з даної таблиці відбувається прив’язка типу судна до еталонного маршруту, для якого він призначений.

Таблиця 2.6 – Атрибути таблиці «Ship_types»

№ з/п	Назва атрибуту	Тип даних	Опис
1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Type	VARCHAR	Назва типу суден

Таблиця «Best_routes» призначена для зберігання еталонних маршрутів з їх описом, контрольними точками, описом даного маршруту, акваторії, до якої він належить, кодову назву маршруту та математичну залежність, що відображає рух судна за маршрутом. Згідно записів даної таблиці будуть проводитись основні дослідження AIC.

Таблиця 2.7 – Атрибути таблиці «Best_routes»

№ з/п	Назва атрибуту	Тип даних	Опис
1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Name	VARCHAR	Кодова назва маршруту
3.	Water_area_id	INT	Ідентифікатор акваторії, до якої даний маршрут належить
4.	User_id	INT	Ідентифікатор користувача, який створив даний маршрут
5.	Dependency_id	INT	Ідентифікатор математичної залежності, що відображає маршрут
6.	Points	JSON	Масив контрольних точок маршруту
7.	Created_at	DATE	Дата створення маршруту

Таблиця «Observations» призначена для переліку спостережень з датою їх проведення, ідентифікатором користувача, який дане спостереження проводив, та ідентифікатором еталонного маршруту, відносно якого проводиться дослідження. Це є однією із основних таблиць, з якими проводиться робота автоматизованої інформаційної системи.

Таблиця 2.8 – Атрибути таблиці «Observations»

№ з/п	**Назва атрибуту**	**Тип даних**	**Опис**
1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Name	VARCHAR	Кодова назва спостереження
3.	Route_id	INT	Ідентифікатор еталонного маршруту, з яким проводилося порівняння
4.	User_id	INT	Ідентифікатор користувача, який робив дане спостереження
5.	Points	JSON	Контрольні точки маршруту
6.	Conclusion	VARCHAR	Висновок щодо необхідності реагування
7.	Created_at	DATE	Дата проведення спостереження

Таблиця «Dependencies» призначена для зберігання типів математичних залежностей, за якими можна найточніше відобразити маршрут судна, як за еталонними маршрутами, та і за маршрутами, над якими проводяться спостереження.

Таблиця 2.9 – Атрибути таблиці «Dependencies»

№ з/п	**Назва атрибуту**	**Тип даних**	**Опис**

1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Dependency	VARCHAR	Назва математичної залежності

Таблиця «Best-routes-ships_pivot» є розвідною таблицею між таблицями «Best_routes» та «Ship_types», відповідно у ній є зовнішні ключі, що вказують на відповідні поля даних таблиць. За даними цієї таблиці відбувається зв’язка еталонного маршруту із типами суден, для яких вони призначені.

Таблиця 2.10 – Атрибути таблиці «Best-routes-ships_pivot»

№ з/п	**Назва атрибуту**	**Тип даних**	**Опис**
1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці
2.	Route_id	INT	Зовнішній ключ, що вказує на еталонний маршрут
3.	Ship_type_id	INT	Зовнішній ключ, що вказує на тип судна, для якого даний маршрут призначений

Таблиця «Messages» призначена для забезпечення комунікації між оператором ЦУС регіонального управління Морської охорони та оператором Головного ЦУС Адміністрації Держприкордонслужби.

Таблиця 2.11 – Атрибути таблиці «Messages»

№ з/п	**Назва атрибуту**	**Тип даних**	**Опис**
1.	ID	INT	Первинний ключ для однозначної ідентифікації записів у таблиці

2.	Sender_id	INT	Зовнішній ключ користувача, який надіслав повідомлення
3.	Receiver_id	INT	Зовнішній ключ користувача, який отримав повідомлення
4.	Send_at	DATETIME	Дата та час надсилання повідомлення

З урахуванням положень п. 2.1.2 можна зробити наступні висновки.

1. Для бізнес-процесу «Авторизація» будуть застосовуватись таблиці «Users» та «User_types».

2. Для бізнес-процесу «Робота з реєстром акваторій, в межах яких проводяться спостереження» будуть застосовуватись таблиці «Water_areas» та «Water_areas_types» бази даних.

3. Для бізнес-процесу «Робота з реєстром еталонних маршрутів суден» застосовуватимуться таблиці «Best_routes» та «Best-routes-ships_pivot» бази даних.

4. Для бізнес-процесу «Робота з модулем оцінки маршрутів суден» будуть застосовуватись таблиці «Best_routes», «Observations», «Dependencies» та «Ship_types» бази даних.

5. Для бізнес-процесу «Робота з реєстром спостережень» застосовуватиметься таблиця «Observations» бази даних.

6. Для бізнес-процесу «Попередня оцінка загроз і ризиків щодо можливого порушення прикордонного законодавства» застосовуватимуться таблиці «Best_routes», «Best-routes-ships_pivot», «Water_areas», «Observations», «Dependencies» бази даних.

7. Для бізнес-процесу «Взаємодія з Головним ЦУС Адміністрації Держприкордонслужби» застосовуватимуться таблиці «Messages» та «Observations» бази даних.

8. Для бізнес-процесу «Оцінка загроз і ризиків щодо можливого порушення прикордонного законодавства» застосовуватимуться таблиці

«Best_routes», «Best-routes-ships_pivot», «Water_areas», «Observations», «Dependencies» бази даних.

9. Для бізнес-процесу «Формування висновків за результатами оцінки загроз і ризиків» застосовуватимуться таблиці «Observations» та «Best_routes» бази даних.

10. Для бізнес-процесу «Взаємодія з ЦУС регіонального управління Морської охорони» застосовуватимуться таблиці «Messages» та «Observations» бази даних.

Отже, представлена база даних відповідає аспектаим функціонування досліджуваної автоматизованої інформаційної системи.

3 Програмна реалізація

3.1 Структура і функціональне призначення модулів системи, їх взаємозв'язок

Для розробки автоматизованої інформаційної системи використовується фреймворк ASP.NET MVC. Суть шаблону MVC полягає в розділенні програми на моделі, контролери та представлення. Контролер реалізовує зв'язок між моделями і представленнями та приймає вхідні дані, які надходять від користувача.

Даний програмний продукт включає в собі наступні контроллери:

1. AuthController – призначений для авторизації та реєстрації користувача в системі, та розділяє користувачів за типами.

2. HomeController – призначений для відображення головної сторінки інформаційної системи та особистого електронного кабінету користувача.

3. BestRouteController – призначений для керування еталонними маршрутами, їх створення, редагування, перегляду та видалення.

4. WaterAreaController – призначений для керування акваторіям, їх додавання до системи, редагування інформації про них, передляду детальної інформації про кожну з них та видалення акваторій з інформаційної системи.

5. ObservationController – призначений для створення спостережень, розрахунку відповідності маршрутів відносно еталонних, збереження даних про спостереження. Крім того, якщо користувач не впевнений щодо висновку отриманого відносно певного спостереження, він може направити дане спостереження на розгляд старшої за званням чи більш компетентної людини. При задовільному результаті розрахунків щодо відповідності маршрутів, досліджуваний маршрут можна перевести до класу еталонних.

Крім цього програмний продукт включає в собі наступні моделі:

1. User – є моделлю, що відповідає за бізнес-логіку, пов'язану із інформацією про користувачів інформаційної системи.

2. UserType – є моделлю, яка відповідає за визначення та розрізнення користувачів за типами.

3. WaterArea – є моделлю, що відповідає за бізнес логіку, пов’язану з даними про акваторії, що знаходяться у реєстрі інформаційної системи.

4. WaterAreaType – є моделлю, що відповідає за бізнес-логіку, пов’язану з розподіленням акваторій за типами.

5. ShipType - є моделлю, що відповідає за бізнес-логіку, пов’язану з розподіленням кораблів за типами.

6. TransferObject – є моделлю, що відповідає за передачу даних між певними окремими відображеннями при створенні спостереження користувачем.

7. BestRoute – є моделлю, що пов’язана з бізнес-логікою відповідальною за збереження та керування даними про еталонні маршрути кораблів.

8. Dependency – є моделлю, що відповідає за бізнес-логіку, пов’язану з збереженням інформації щодо математичних залежностей, які використовуються для побудови графіків маршрутів при спостереженнях.

9. Observation – є моделлю, що відповідає за бізнес-логіку, пов’язану з створенням спостережень.

10. Context – є базовою моделлю, що відповідає за підключення системи до бази даних.

Крім цього, дана інформаційна система включає в собі наступні основні відображення:

1. Index – відповідає за відображення головної сторінки автоматизованої інформаційної системи.

2. Observation.Index – відповідає за створення спостереження та відображення результату розрахунку відповідності досліджуваних маршрутів.

3. WaterArea.Index – відповідає за відображення даних про усі акваторії зареєстровані у інформаційній системі.

4. WaterArea.Show – відповідає за відображення детальної інформації про обрану користувачем акваторію.

5. BestRoute.Index – відповідає за відображення даних про усі еталонні маршрути, зареєстровані в системі.

6. BestRoute.Show – відповідає за відображення детальної інформації про обраний еталонний маршрут.

7. Observation.Investigation – відповідає за відображення даних про усі спостереження, що потребують додаткового дослідження.

8. Observation.InvestigationShow – відповідає за відображення детальної інформації про обране спостереження, що потребує додаткового дослідження.

На рисунку 3.1 зображено діаграму класів розробленого веб-сайту.

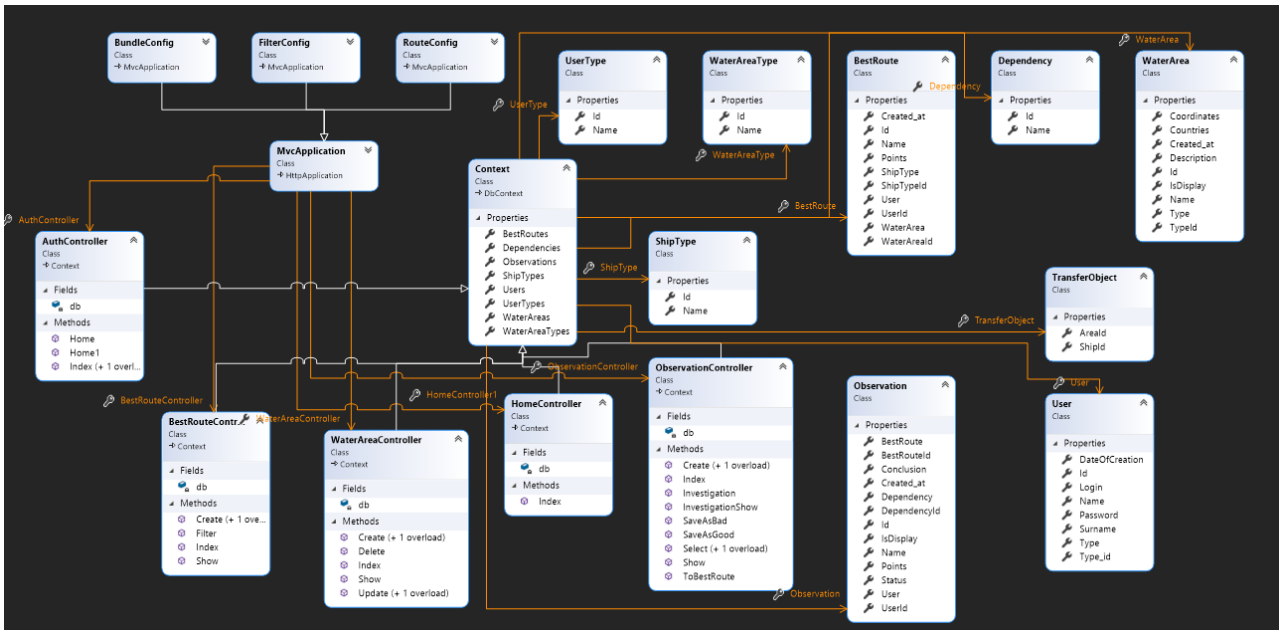


Рисунок 3.1 – діаграма класів веб-застосунку

Також у програмному продукті використовується набір інших компонентів для кращої взаємодії з системою, до прикладу можна віднести Bootstrap та бібліотеку для роботи із JavaScript – JQuery.

Наведений список моделей, представлень та контролерів забезпечить функціонал інформаційної системи. Подібне розділення програмного продукту на три даних компоненти дозволить значно спростити розроку веб-застосунку.

3.2 Розробка програмних модулів

Найважливішими частинами алгоритмів програмного коду для оператора автоматизованої інформаційної системи є:

- авторизація;
- робота з реєстром акваторій;
- робота з реєстром еталонних маршрутів;
- можливість створювати спостереження;
- можливість направляти спостереження на додаткове дослідження;
- можливість проводити додаткові дослідження;
- можливість робити висновки, щодо відповідності досліджуваного маршруту до еталонного;
- можливість зберігати досліджуваний маршрут до реєстру еталонних.

При вході користувача у інформаційну систему в першу чергу йому відображається головна сторінка АІС.

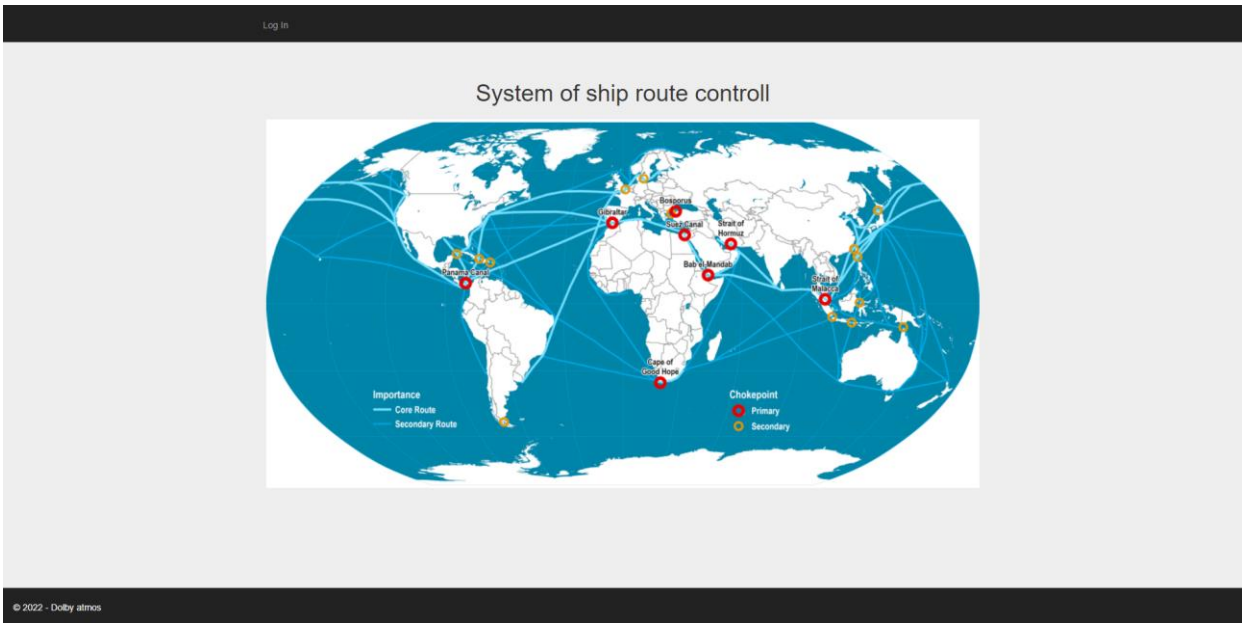


Рисунок 3.2 – головна сторінка інформаційної системи

Єдине, що може зробити користувач на даній сторінці це перейти до етапу авторизації, обравши відповідний пункт у меню.

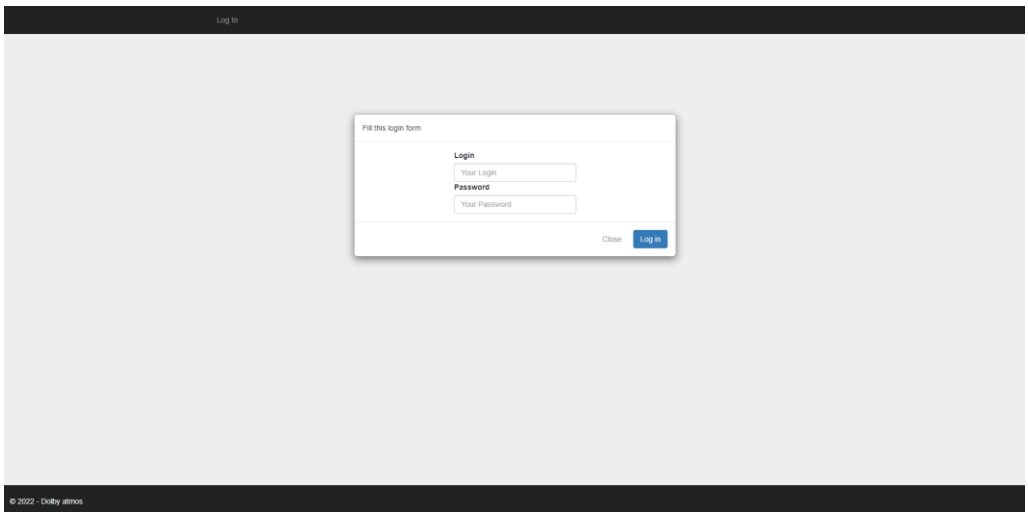


Рисунок 3.3 – сторінка авторизації користувача

Користувачеві необхідно заповнити форму для подальшої роботи з системою. Якщо користувач зоповнить форму невірними чи не валідними даними, йому буде відображена сторінка в відповідним описом проблеми.

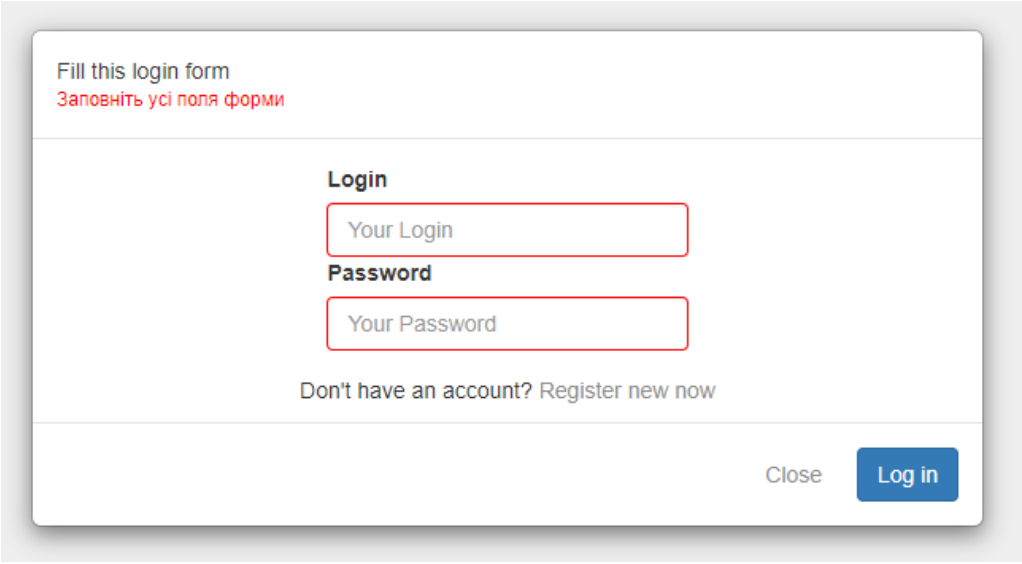


Рисунок 3.4 – відображення помилки при авторизації

Якщо ж користувач заповнить форму правильними даними, які відповідають логіну та паролю, призначеним для авторизації саме цього користувача, йому буде відображено повідомлення про успішну авторизацію.

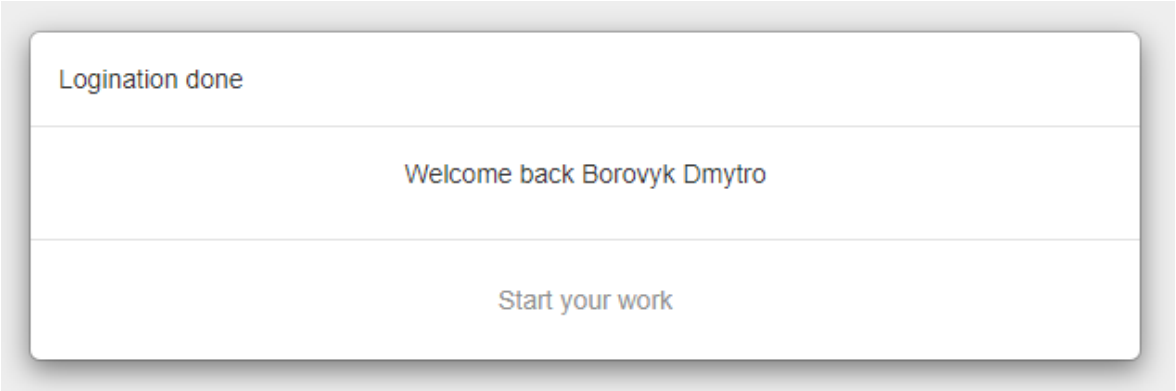


Рисунок 3.5 – відображення повідомлення про успішну авторизацію

В залежності від типу користувача, що авторизується йому буде відображатись особистий кабінет з різним рівнем наповненості головного меню. Особистий електронний кабінет користувача ,що належить до регіонального ЦУС, можна побачити на рисунку 3.6.

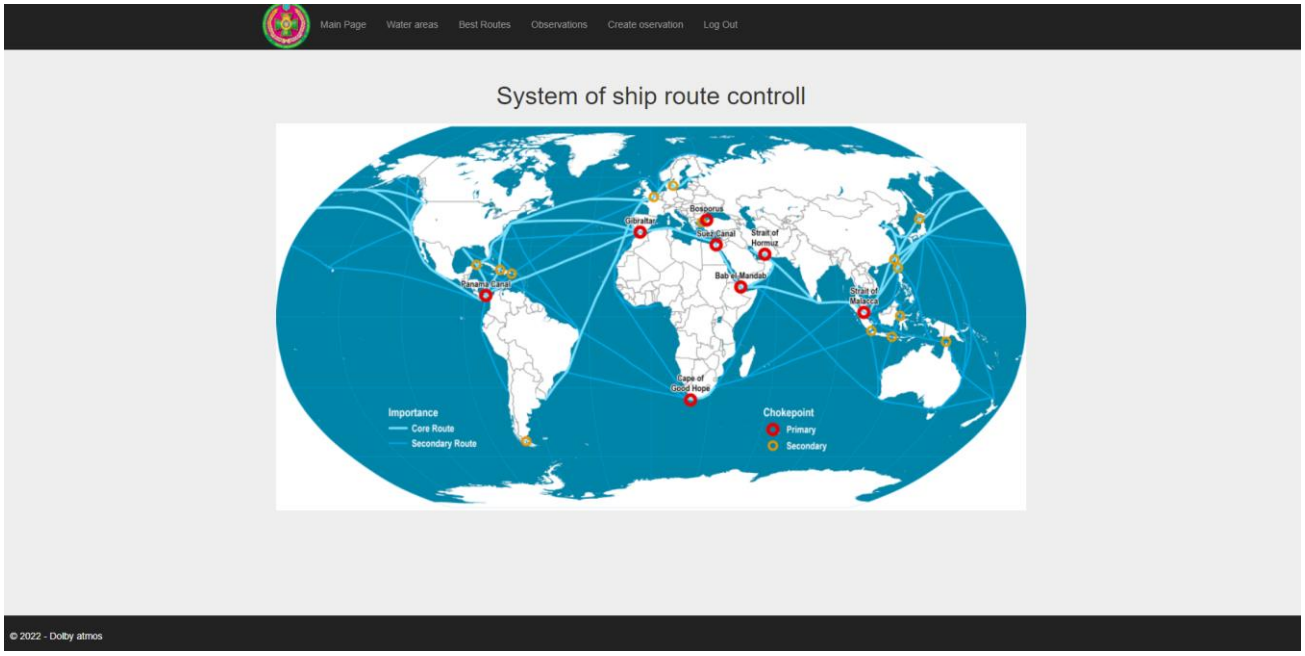


Рисунок 3.6 – особистий кабінет користувача регіонального ЦУС

Якщо ж користувач належить до головного ЦУС, то йому відобразиться така ж сторінка, проте у головному меню з’являться додаткові пункти. Головне меню користувача головного ЦУС можна побачити на рисунку 3.7.

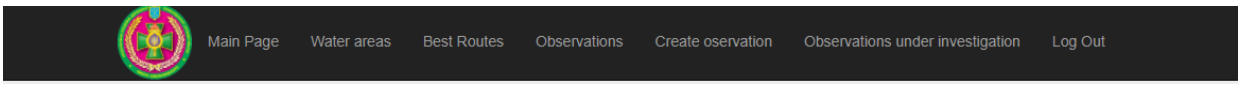


Рисунок 3.7 – головне меню користувача головного ЦУС

Для роботи із реєстром акваторій, користувачу необхідно обрати відповідний пункт у головному меню. Після цього користувачеві буде відображено перелік усіх акваторій, зареєстрованих у автоматизованій інформаційній системі.

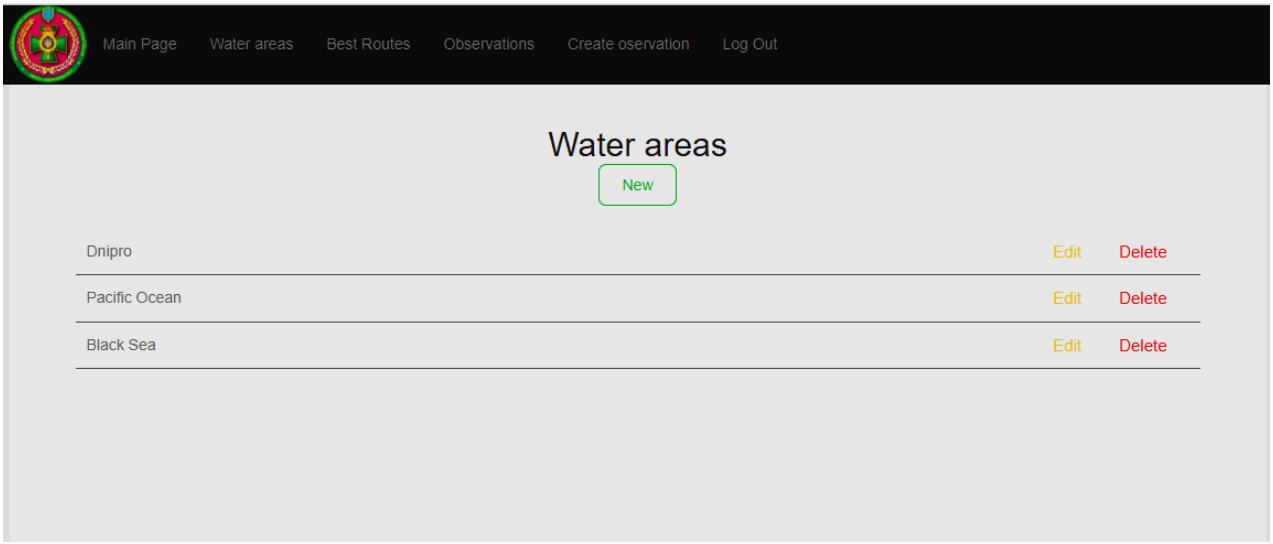


Рисунок 3.8 – сторінка відображення усіх зареєстрованих акваторій

Користувач може додати нову акваторій, переглянути інформацію про вже існуючу акваторію, відредагувати інформацію про акваторію чи видалити акваторію з реєстру, обравши відповідну дію. Для перегляду детальної інформації про акваторію, користувач повинен навести курсор на її назву та натиснути на неї, після чого користувачеві буде відображено сторінку із детальною інформацією про неї.

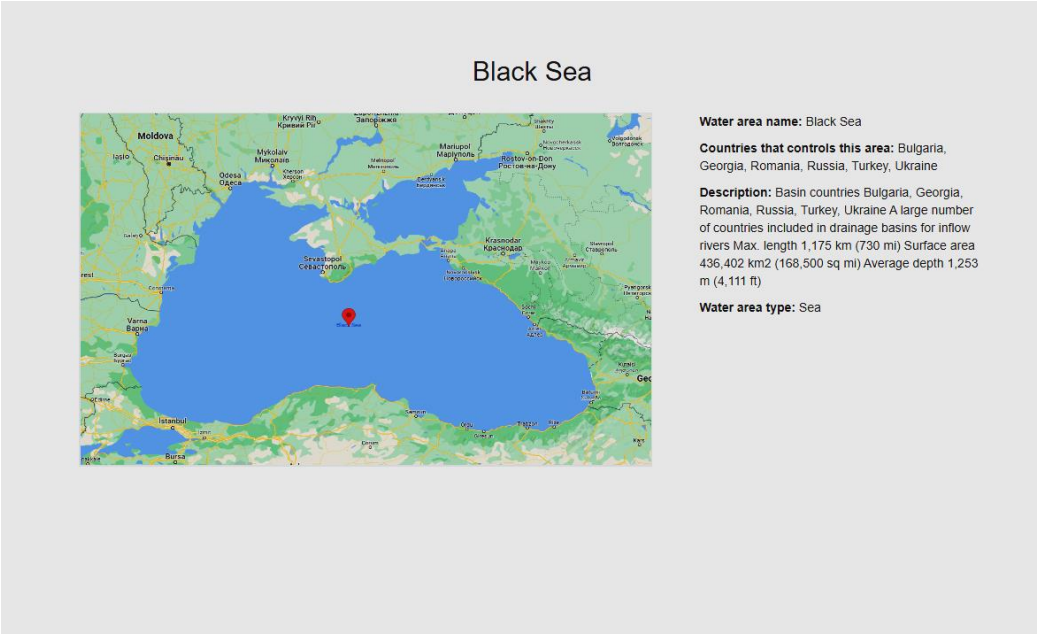


Рисунок 3.9 – сторінка відображення детальної інформації про акваторію

Для додавання чи редагування інформації про акваторію, користувач повинен обрати відповідну дію у меню, після чого йому буде відображено форму додавання чи редагування інформації про акваторію.

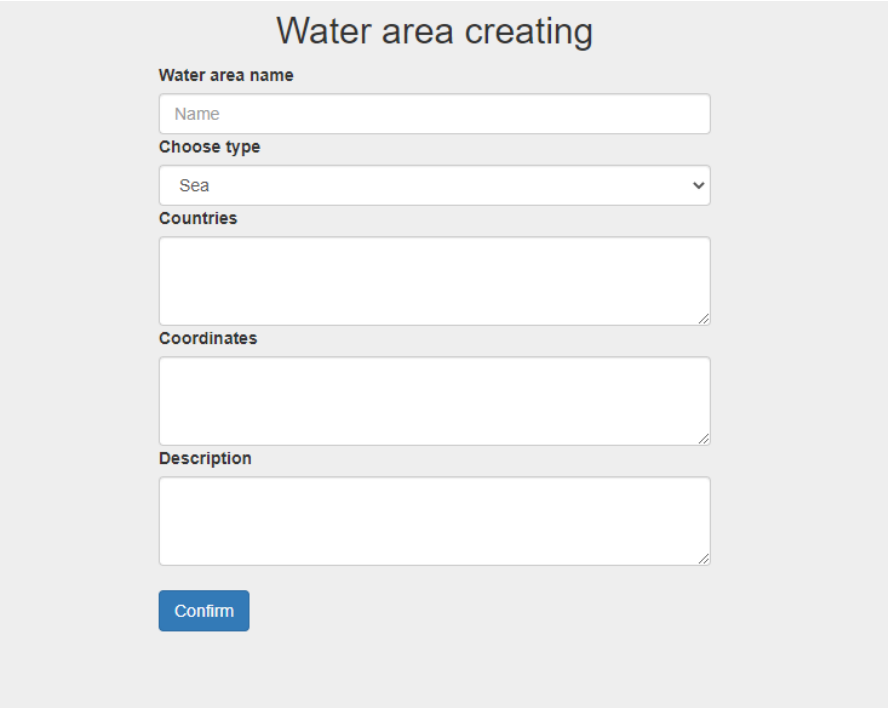


Рисунок 3.10 – форма додавання нової акваторії

Для перегляду реєстру еталонних маршрутів, користувачеві необхідно обрати відповідний пункт у головному меню, після натискання на нього, йому буде відображено перелік усіх еталонних маршрутів.

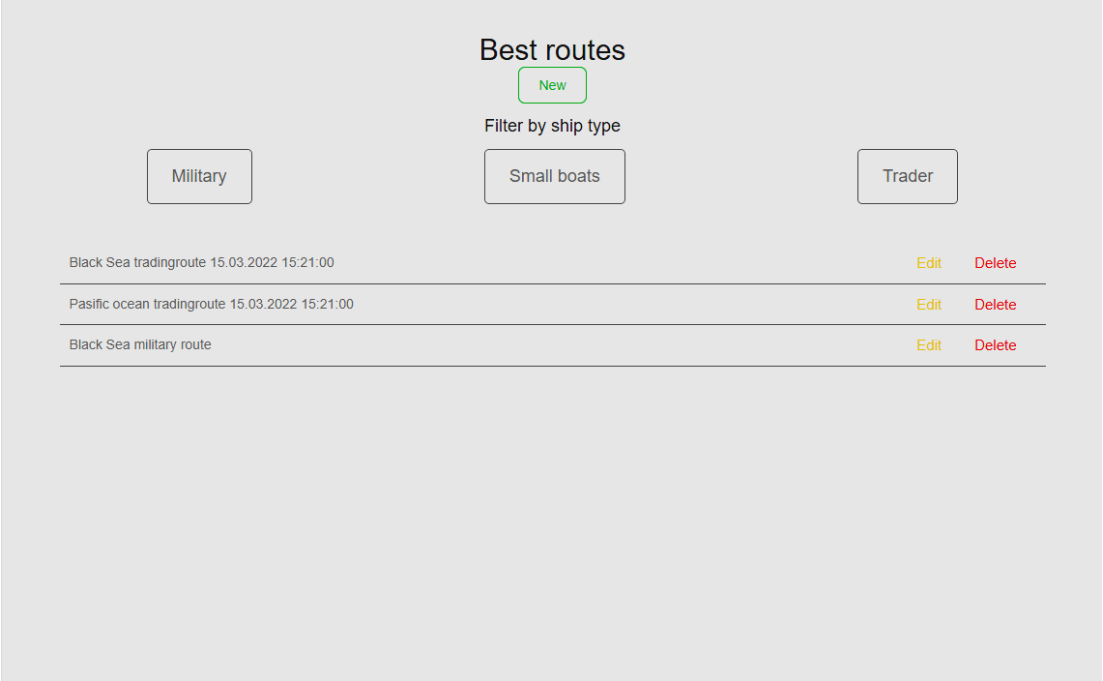


Рисунок 3.11 – сторінка відображення даних про усі еталонні маршрути

Користувач може провести фільтрацію маршрутів за типами кораблів для яких вони призначені. Також користувач може додати новий маршрут, переглянути детальну інформацію про маршрут, відредагувати дані про обраний маршрут та видалити обраний маршрут. Обравши пункт перегляду детальної інформації користувачеві відображається відповідна сторінка із інформацією про даний маршрут, положення його акваторії на карті та графіку, що відображає його за контрольними точками та звичайною лінійною апроксимацією в якості математичної залежності. На рисунку 3.12 можна побачити сторінку детальної інформації про еталонний маршрут.

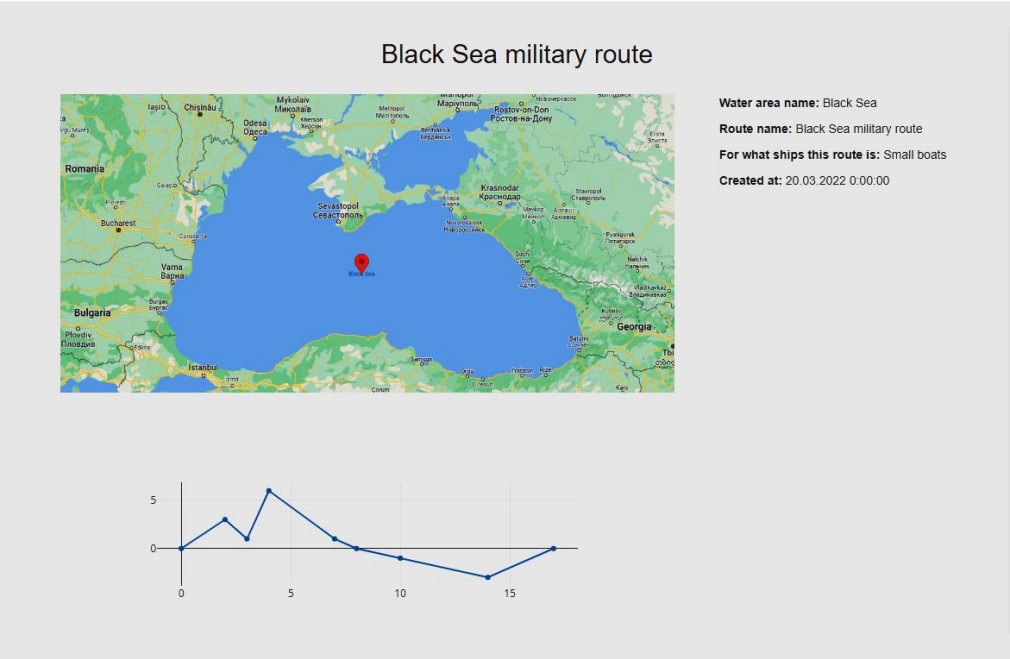


Рисунок 3.12 – сторінка відображення детальної інформації про еталонний маршрут

Для додавання чи редагування інформації про еталонні маршрути, користувач повинен обрати відповідну дію у меню, після чого йому буде відображено форму, яку користувачеві буде необхідно заповнити.

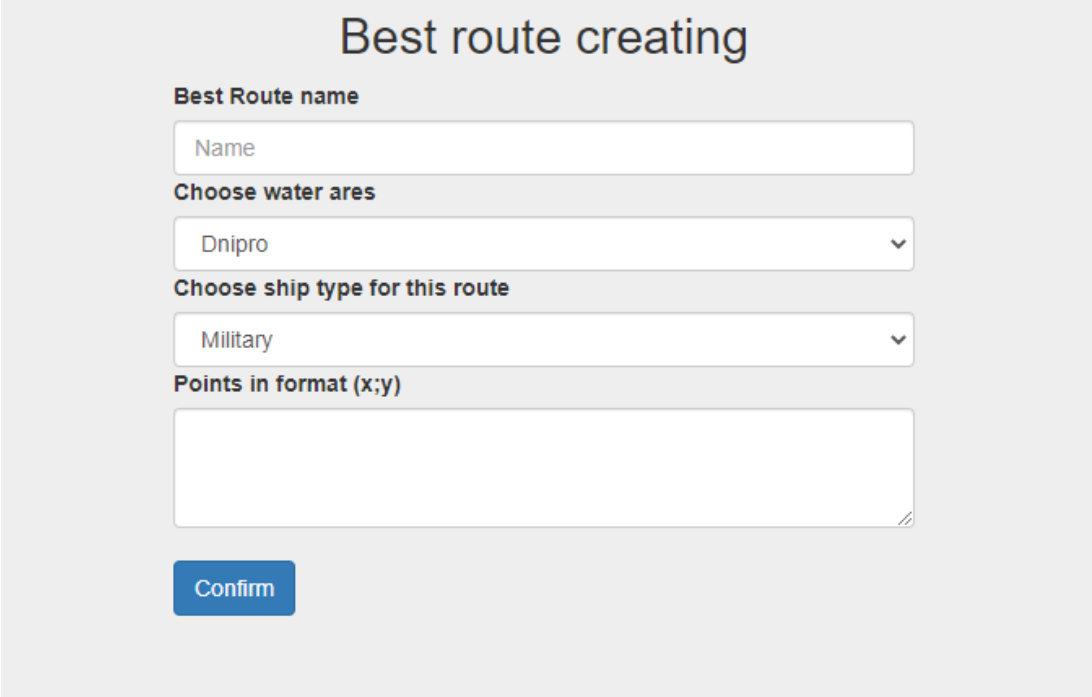


Рисунок 3.13 – форма створення нового еталонного маршруту

Для створення спостереження користувачеві необхідно обрати відповідний пункт у головному меню, після чого йому буде відображено поле вибору акваторії та маршруту за якими буде проводитися спостереження.

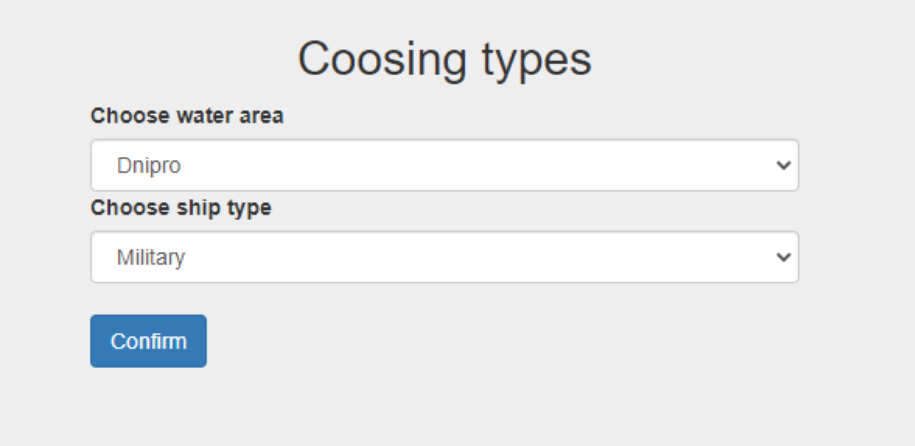


Рисунок 3.14 – поле вибору даних для створення спостереження

Після того, як користувач обрав вхідні данні, йому буде відображено сторінку створення спостереження, на які буде відображено знімок акваторії, графік, що відображає траекторію еталонного маршруту та форму для проведення дослідження.

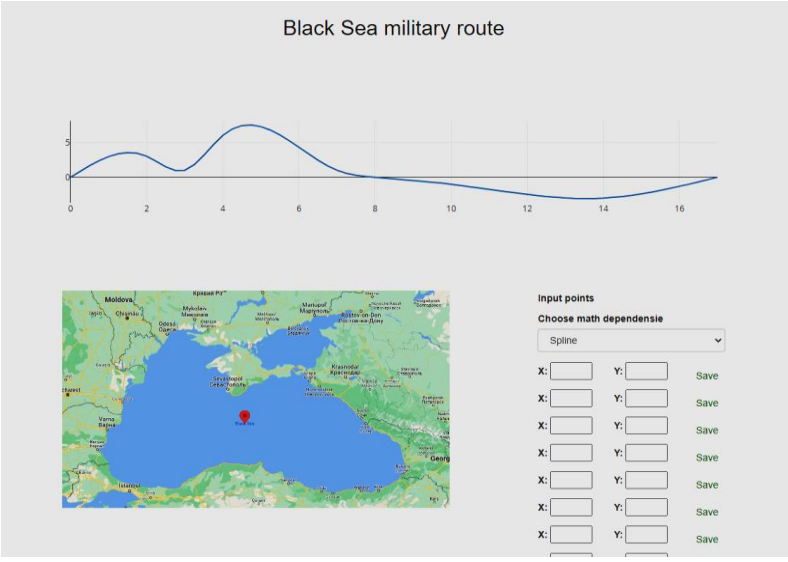


Рисунок 3.15 – сторінка створення спостереження

Заповнивши поля форми значеннями, користувач може побачити графік, що відображає траекторію досліджуваного маршруту корабля, за обраною математичною залежністю та висновок, щодо оцінки небезпеки та подібності даних маршрутів.

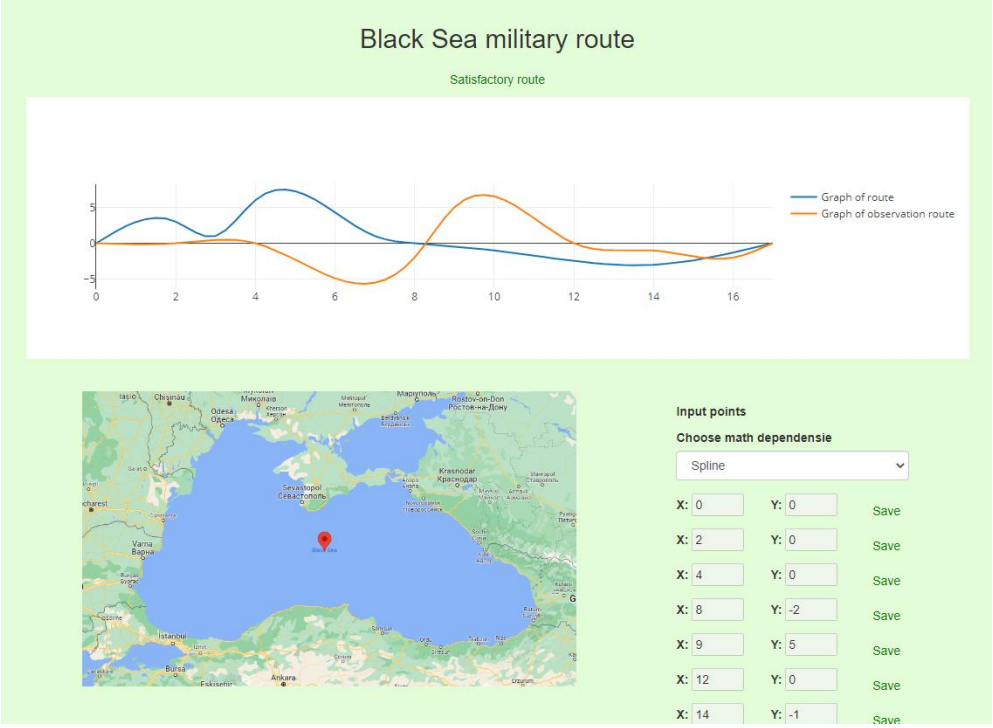


Рисунок 3.16 – результат створеного спостереження із висновком

Після отримання висновку користувач може визначити, чи зберегти дане спостереження із отриманим результатом, чи направити спостереження на додаткове дослідження. Після збереження спостереження, його можна побачити на сторінці усіх спостережень.

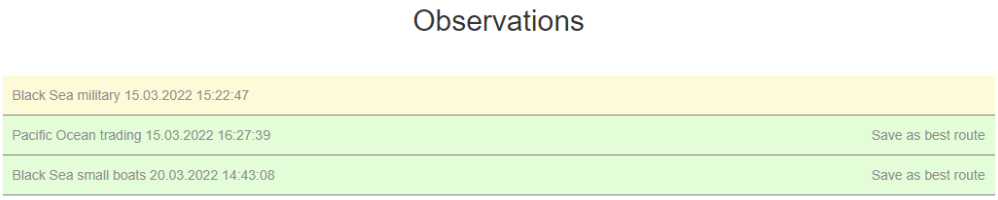


Рисунок 3.17 – сторінка відображення останніх досліджень

Якщо спостереження отримало задовільний результат, то досліджуваний маршрут користувач може перевести у клас еталонних маршрутів, натиснувши відповідну кнопку на екрані.

Переглянути перелік маршрутів, що потребують додаткового перегляду, можуть тільки працівники головного ЦУС, обравши відповідний пункт меню.

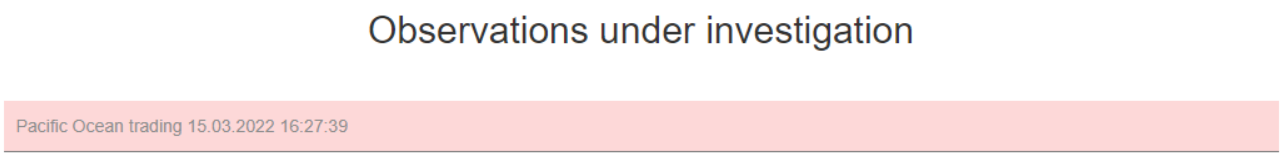


Рисунок 3.18 – сторінка відображення спостережень, що потребують перегляду

Користувач може обрати спостереження, яке він хоче переглянути, натиснувши на його назву.

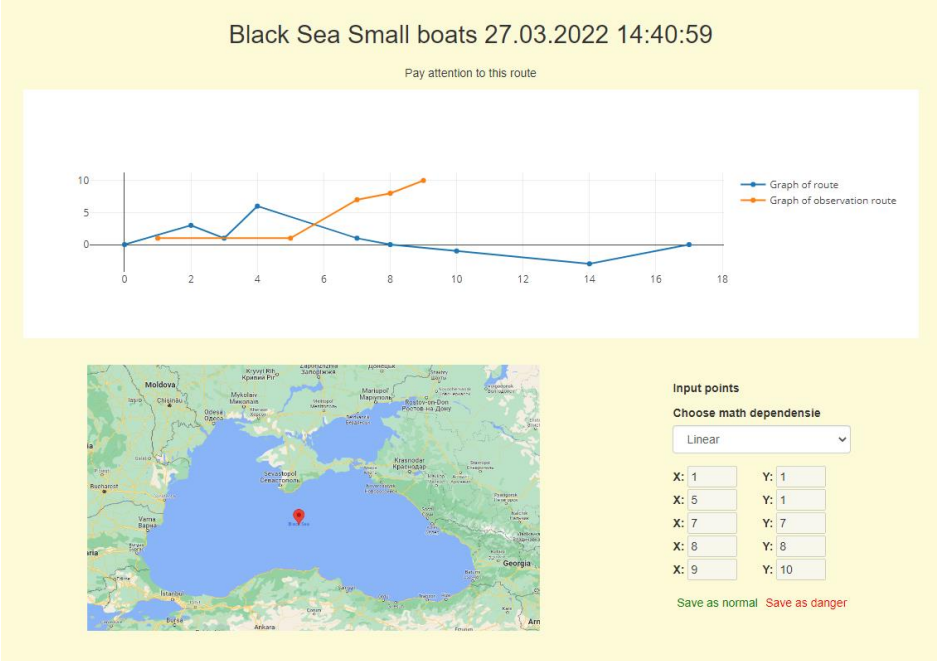


Рисунок 3.19 – сторінка додаткового перегляду спостереження

Користувач, що проводить додаткове дослідження, може провести аналіз та визначити чи є задовільним маршрут, чи навпаки небезпечним.

3.3 Тестування інформаційної системи

Для перевірки роботи інформаційної системи було проведено його функціональне та модульне тестування. Згідно поставленому завданню, застосунок має групи функцій для роботи з реєстрами акваторій, еталонних маршрутів, спостережень. Дані які отримують функції, повинні бути перевірені для запобігання виникнення помилок і забезпечення коректного функціонування як АІС, так і БД. Програмний продукт зобов'язаний коректно виводити інформацію з бази даних, мати зрозумілий користувацький інтерфейс, у випадку виникнення помилок, повинен відображати зрозуміле попередження.

Першим модулем для тестування виступає функція авторизації користувача. Дана функція слугує для того, щоб оператор АІС міг отримати доступ до його електронного кабінету. Методом, що забезпечує тестування модуля авторизації є LoginTest. Лістинг коду даного методу можна побачити на рисунку 3.20.

```
[TestMethod()]
0 references
public void LoginTest()
{
    AuthController obj = new AuthController();
    User u = new User();
    u.Login = "TopBor";
    u.Password = "ebidod90";

    ViewResult exp = obj.Index(u.Login, u.Password) as ViewResult;
    Assert.IsNotNull(exp);
}
```

Рисунок 3.20 – лістинг методу LoginTest

Наступною функцією для тестування виступає функція, що відповідвє за додавання нової акваторії до реєстру акваторій. Дана функція слугує для того, щоб оператор АІС міг використовувати власний електронний кабінет, для керування реєстром акваторій. Функцією, що забезпечує тестування даного

модуля є WaterAreaCreateTest. Лістинг коду даного методу можна побачити на рисунку 3.21.

```
[TestMethod()]
0 references
public void WaterAreaCreateTest()
{
    WaterAreaController obj = new WaterAreaController();
    WaterArea area = new WaterArea();
    area.Countries = "test countries";
    area.Description = "test description";
    area.Name = "test";
    area.IsDisplay = 0;
    area.TypeId = 3;
    area.Countries = "test";
    area.Created_at = DateTime.Now;

    ActionResult exp = obj.Create(area) as ActionResult;
    Assert.IsNotNull(exp);
}
```

Рисунок 3.21 – лістинг методу WaterAreaCreateTest

Наступним модулем для тестування виступає функція, що відповідає за проведення додаткового перегляду спостереження. Дана функція слугує для того, щоб більш компетентна людина могла дати свою експертну оцінку, щодо ситуації. Методом, що забезпечує тестування даного модуля є InvestigationShowTest. Лістинг коду даного методу можна побачити на рисунку 3.22.

```
[TestMethod()]
0 references
public void InvestigationShowTest()
{
    ObservationController obj = new ObservationController();

    int id = 8;
    ActionResult exp = obj.InvestigationShow(id) as ActionResult;
    Assert.IsNotNull(exp);
}
```

Рисунок 3.22 – лістинг методу InvestigationShowTest

Останім модулем для тестування виступає функція, що відповідає за створення висновку щодо додаткового перегляду спостереження. Дана функція слугує для того, щоб оператор AIC міг, використовуючи власний електроний

кабінет для визначення результату додаткового перегляду спостереження. Методом, що забезпечує тестування даного модуля є SaveAsGoodTest. Лістинг коду даного методу можна побачити на рисунку 3.23.

```
[TestMethod()]
0 references
public void SaveAsGoodTest()
{
    ObservationController obj = new ObservationController();
    int id = 8;
    ActionResult exp = obj.SaveAsGood(id) as ActionResult;
    Assert.IsNotNull(exp);
}
```

Рисунок 3.23 – лістинг методу SaveAsGoodTest

Результати модульного тестування можна побачити на рисунку 3.24.

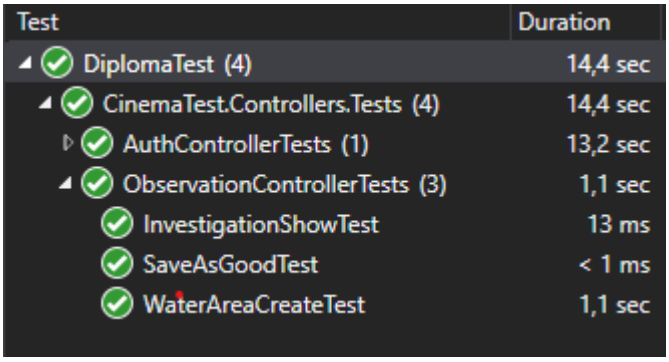


Рисунок 3.24 – результати модульного тестування

Застосунок має визначений набір функцій, а саме:

- перегляд даних;
- додавання даних;
- редагування даних;
- видалення даних;
- створення спостережень;
- авторизація.

Першою функцією для функціонального тестування є авторизація після проходження якої користувач отримає доступ до всіх основних функцій

інформаційної системи. На початку роботи даної функції користувачеві відображається форма авторизації яку необхідно заповнити.

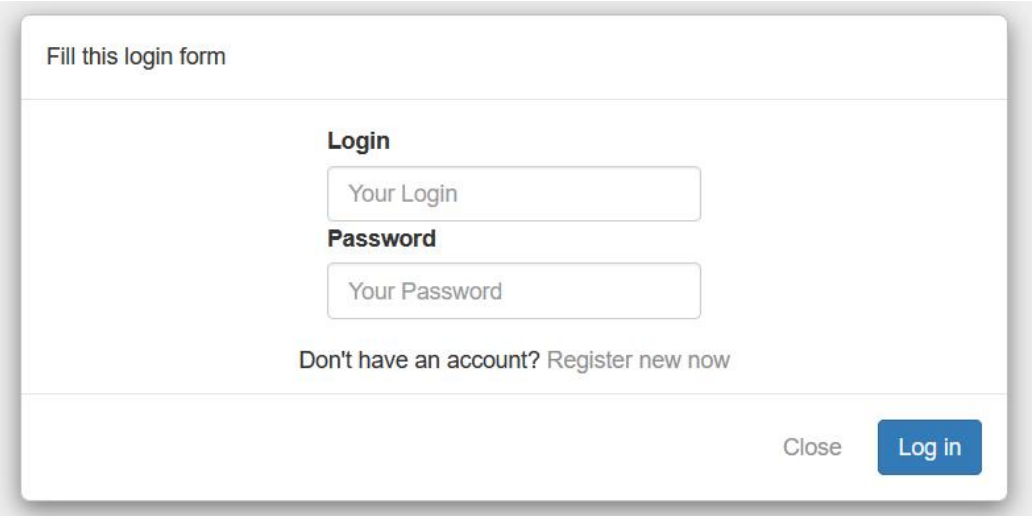


Рисунок 3.25 – форма авторизації

Після заповнення користувачем даної форми відбувається валідація даної форми. Якщо валідація є невдалою, користувачеві виводиться попередження про не коректність введених ним даних.

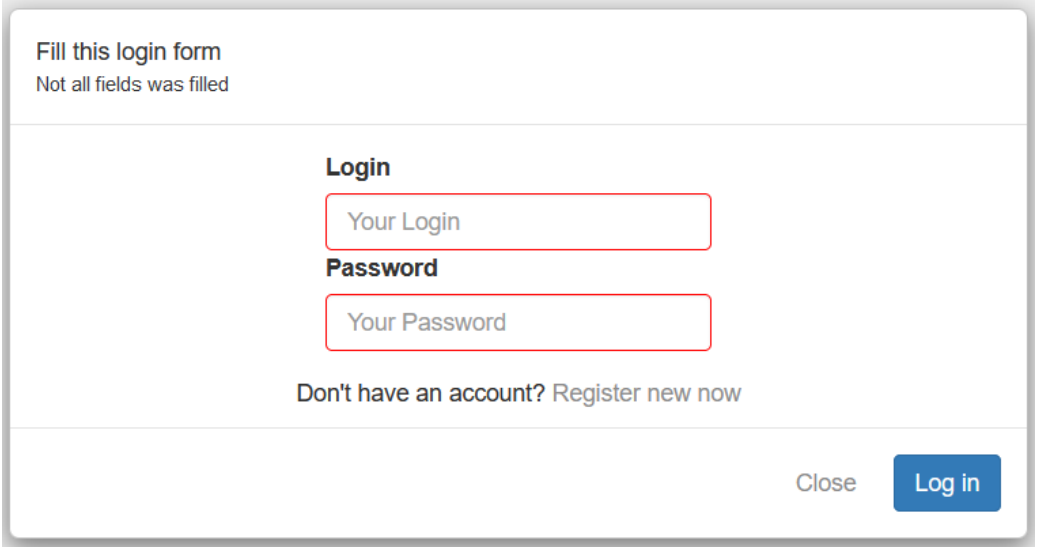


Рисунок 3.26 – попередження про невдалу валідацію

При вдалій валідації проводиться запит до бази даних та відбувається пошук зареєстрованого користувача згідно із даними введеними у поля форми. При вдалій авторизації виводиться вигляд зображений на рисунку 3.27.

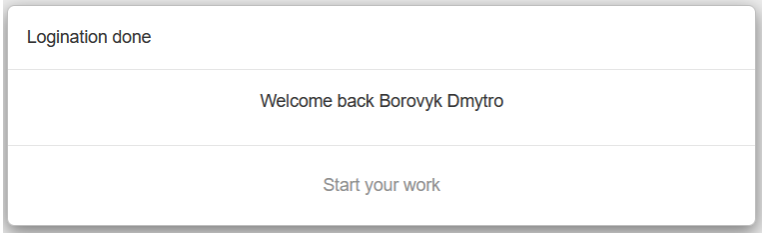


Рисунок 3.27 – вдала авторизація

Однією з основних функцій інформаційної системи є додавання нової акваторії до реєстру інформаційної системи. Дана функція починає свою роботу з відображення форми додавання акваторії, зображеної на рисунку 3.28.

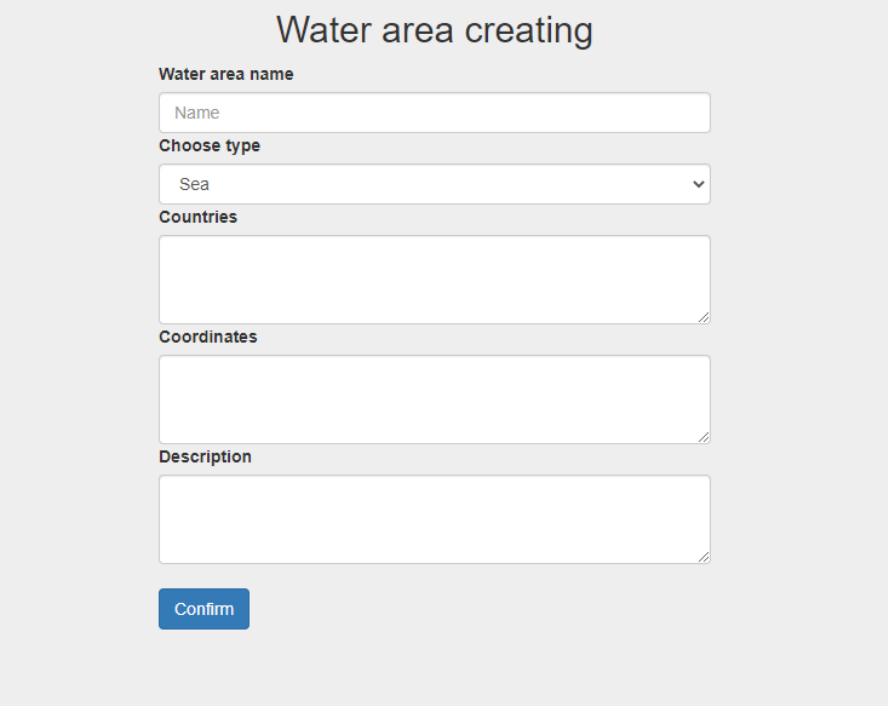


Рисунок 3.28 – форма додавання нової акваторії

Користувачу необхідно коректно заповнити усі поля та написнути на кнопку «Confirm». Після цього відбувається валідація форми та перевірка на

коректність введених даних. Якщо введені дані є некоректними, тоді користувачеві відобразиться попередження зображене на рисунку 3.29.

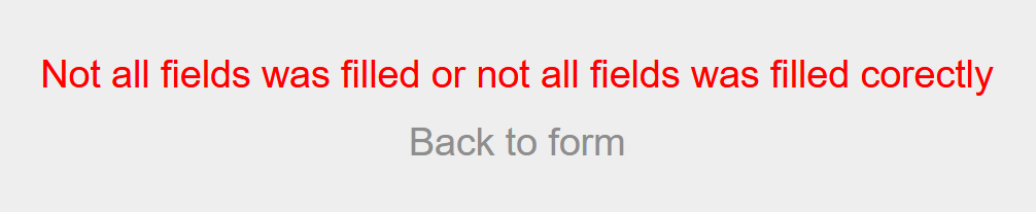


Рисунок 3.29 – попередження про не коректність заповнених даних

Якщо ж введені користувачем дані є коректними та успішно проходять валідацію, виконується запит до бази даних та дані додаються до реєстру.

Аналогічним є створення нового еталонного маршруту, що починається також із відображення користувачеві форми створення еталонного маршруту, зображена на рисунку 3.30.

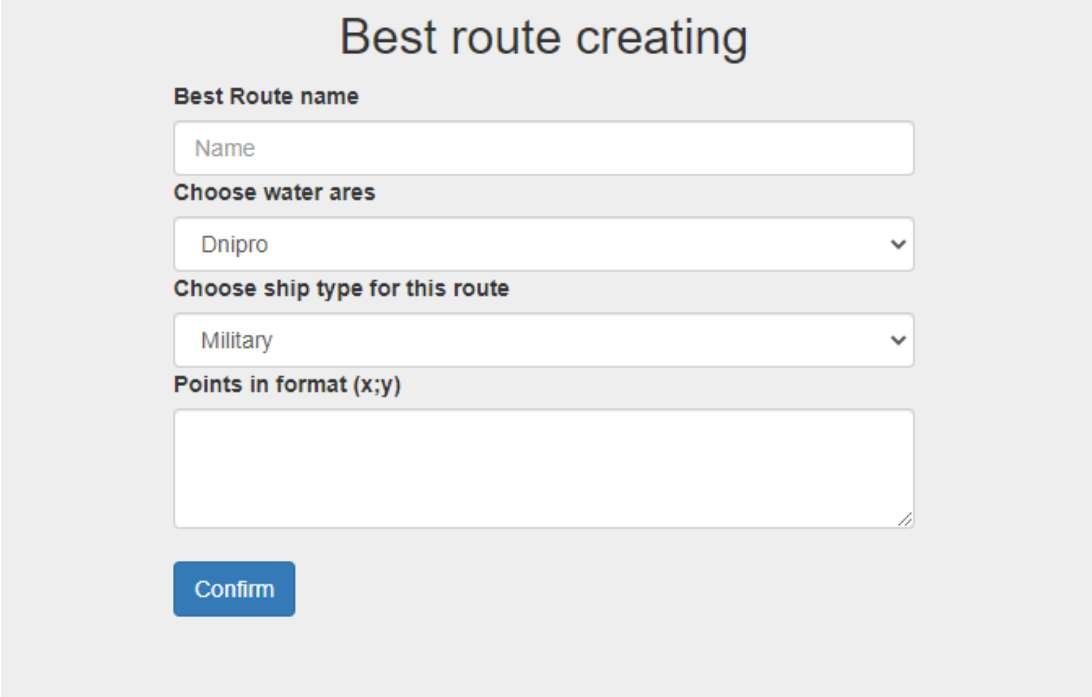


Рисунок 3.30 – форма створення еталонного маршруту

При не валідному заповненні форми, користувачеві відображається аналогічне повідомлення про некоректність заповнення форми, як і при додаванні акваторії.

При вдалій валідації проводиться запит до бази даних та додаються дані щодо нового еталонного маршруту.

3.4 Інструкція користувача

При вході у інформаційну систему, користувачеві відображається головна сторінка AIC, що можна побачити на рисунку 3.31.

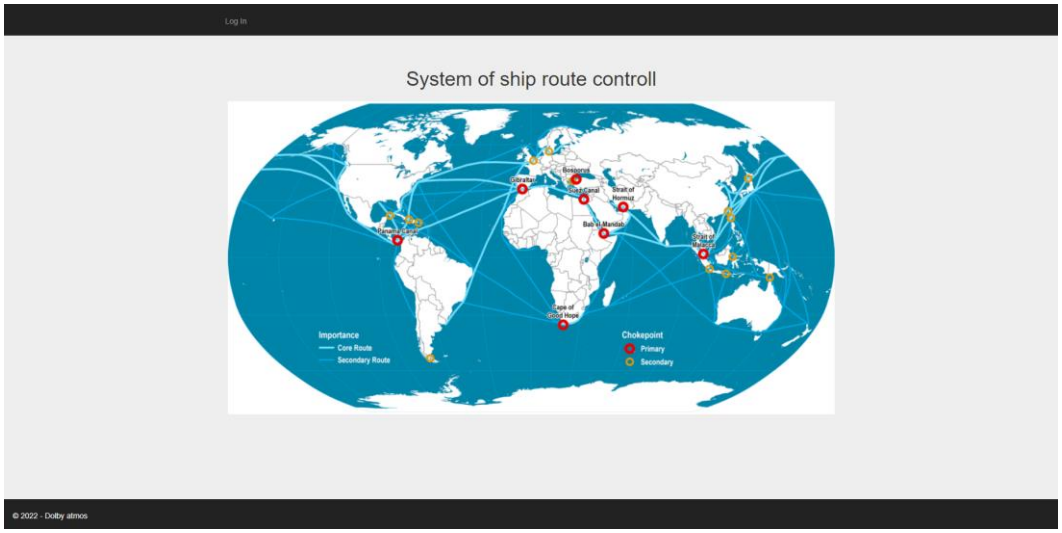


Рисунок 3.31 – головна сторінка інформаційної системи

Далі користувачеві необхідно авторизуватись в системі, для цього він повинен заповнити форму, зображену на рисунку 3.32.

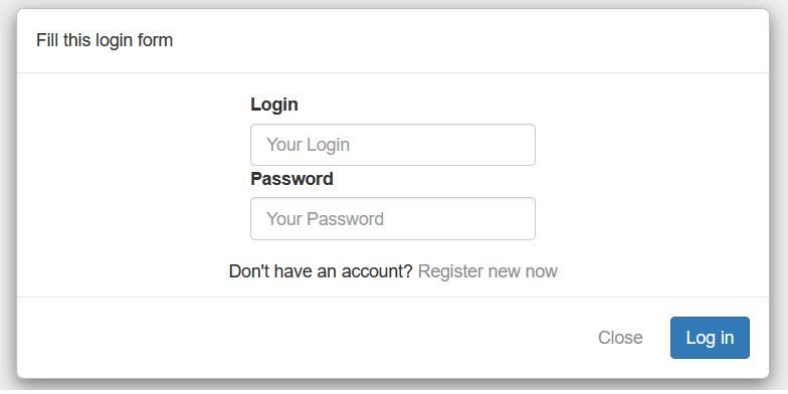


Рисунок 3.32 – форма авторизації

Після успішної авторизації, користувач отримує доступ у особистий електронний кабінет у якому він отримує доступ до усіх функцій інформаційно системи. Користувач може працювати з реєстром акваторій (рисунок 3.33), еталонних маршрутів (рисунок 3.34) та роботу з спостереженями (рисунок 3.35).

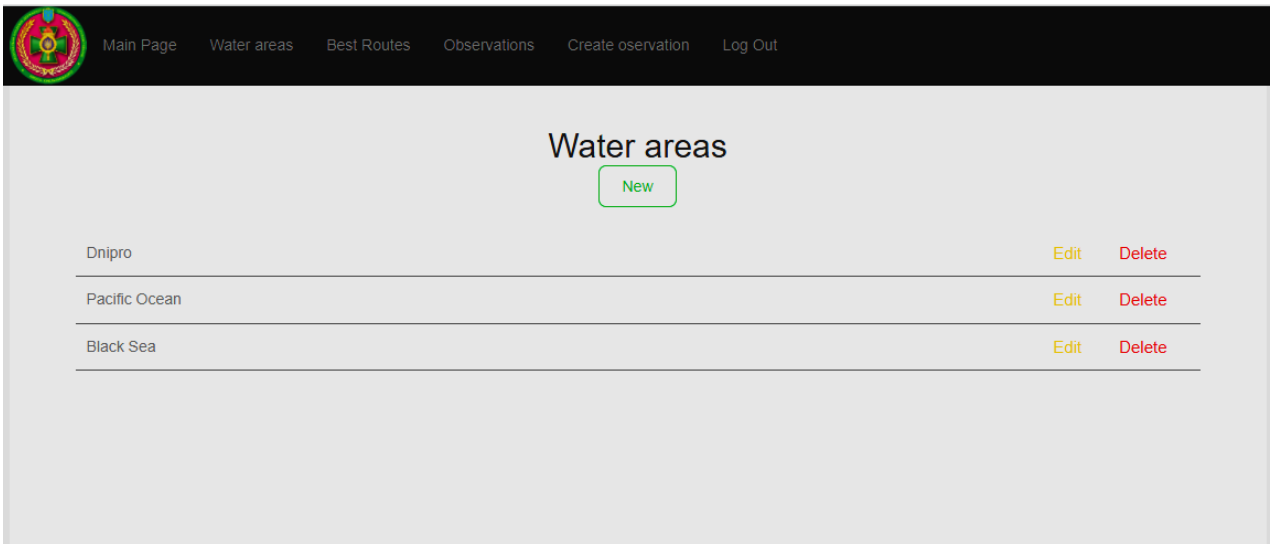


Рисунок 3.33 – реєстр акваторій

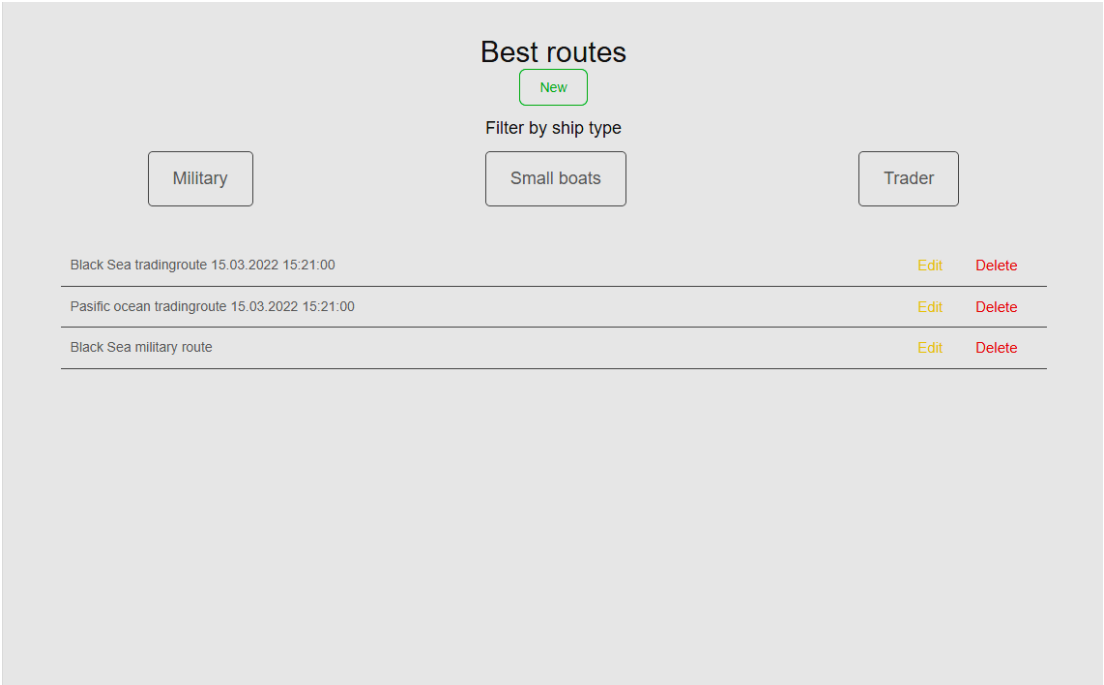


Рисунок 3.34 – реєстр еталонних маршрутів

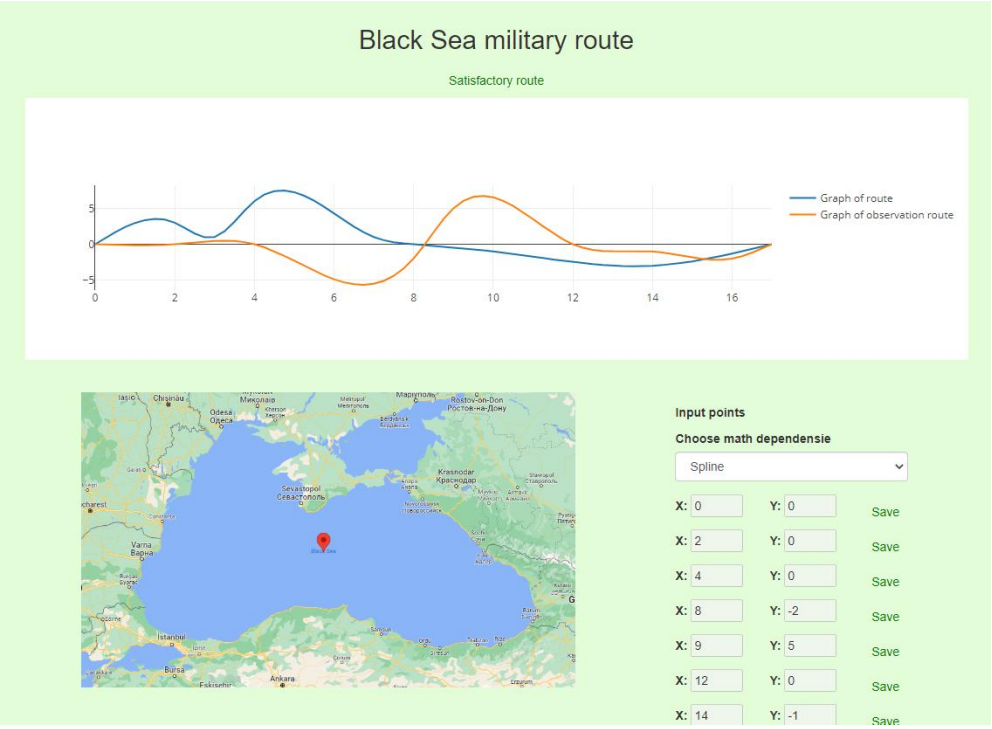


Рисунок 3.35 – робота з стостереженями

Дизайн інформаційної системи є доступним та простим, тому керування інформаційною систему є простим та інтуітивним.

3.5. Вимоги до публікації інформаційної системи

Вимоги до браузера:

– браузер Microsoft Internet Explorer 6.x і вище;
– браузер Mozilla Firefox 3.0 і вище;
– браузер Google Chrome 3.0.195 і вище;
– браузер Safari 3.0 і вище.

Вимоги до провайдера:

– сервер IIS;
– сервер для роботи із базою даних Microsoft SQL Server;
– 512Мб пам’яті на хмарному носії;
– підтримка HTML 5;

- підтримка CSS 3;
- підтримка Bootstrap 4.

Висновки

Під час виконання кваліфікаційної роботи бакалавра був проведений аналіз предметної області, пов’язаної із національною безпекою та запобіганням контрабанді. Також було проаналізовано вже існуюче програмне забезпечення, що зараз використовується у держструктурах. Було поставлено завдання про розробку автоматизованої інформаційної системи порівняння маршрутів кораблів відносно еталонних маршрутів із веб-інтерфейсом та функціями, що необхідні для функціонування даного типу програмного забезпечення.

У результаті виконання кваліфікаційної роботи було розроблено метод для порівняння маршрутів кораблів відносно еталонного маршруту для систем висвітлення надводної обстановки та створення відповідної інформаційної системи для оцінки ступеня відхилення маршруту судна від еталонного маршруту та прийняття рішення щодо необхідності реагування на ймовірне порушення правил прикордонного законодавства. Для розробки програмного продукту було використано мову програмування C#, також платформу, що є фреймворком мови C# – ASP.Net MVC та СКБД – SQL Server.

Розроблена інформаційна система може бути впроваджена для діючих структурах, де програмне забезпечення такого типу є необхідним. Вдосконаливши програмний продукт розширенням функціоналу, й реалізувавши можливість роботи більшої кількості операторів, може зробити систему досить унікальною.

Відповідно до отриманого результату можна зробити висновок, що розроблені метод та автоматизована інформаційна система оцінки відповідності маршрутів кораблів до еталонного маршруту працює вірно та виконує усі вимоги поставлені технічним завданням у повному обсязі.

Список використаної літератури

1. Про Державну прикордонну службу України: Закон України № 6612-IV від 3 квітня 2003 р. // Відомості Верховної Ради. – 2003. – № 27. – С. 208.

2. Стратегія розвитку Державної прикордонної служби: Схвалено розпорядженням Кабінету Міністрів України від 23 листопада 2015 р. № 1189-р.

3. Концепція інтегрованого управління кордонами, схвалена розпорядженням Кабінету Міністрів України від 28 жовтня 2015 р. № 1149: законодавство України, сайт Верховної ради України. – [Електронний ресурс] // Режим доступу:http //zakon3.gov.ua/laws/show/1149-2015-p#n10

4. Мазур В. Ю., Боровик О. В. Методичні основи формування концепції розбудови системи висвітлення надводної обстановки на морській (річковій) ділянці в контексті забезпечення прикордонної безпеки // Сучасні інформаційні технології у сфері безпеки та оборони Вип. № 3(30). – К.: Національний університет оборони України імені Івана Черняховського, Інститут інформаційних технологій, 2017. – С. 137-145.

5. Мазур В. Ю., Боровик О. В. Концепція розбудови єдиної системи висвітлення надводної обстановки на морській (річковій) ділянці в контексті забезпечення прикордонної безпеки // Сучасні інформаційні технології у сфері безпеки та оборони Вип. № 1(31). – К.: Національний університет оборони України імені Івана Черняховського, Інститут інформаційних технологій, 2018. – С. 137-142.

6. Мазур В. Ю., Боровик О. В. Функціональний аналіз варіантів створення єдиної системи висвітлення надводної обстановки на морській (річковій) ділянці в контексті забезпечення прикордонної безпеки // Збірник наукових праць № 4 (74). Серія: Військові та технічні науки. – Хмельницький: Вид. НАДПСУ, 2017. – С. 158-175.

7. Мазур В. Ю., Боровик О. В. Методичні основи оцінки ефективності функціонування єдиної системи висвітлення надводної обстановки на морській

ділянці. – Наука і техніка Повітряних Сил Збройних Сил України № 2 (31). – Харків: ХНУПС, 2018. – С. 182-189.

8. Мазур В. Ю., Боровик О. В. Методика оцінки ефективності функціонування системи висвітлення надводної обстановки на морській ділянці: інформаційні аспекти // Збірник наукових праць № 2 (76). Серія: Військові та технічні науки. – Хмельницький: Вид. НАДПСУ, 2018. – С. 72-86.

9. Боровик О. В., Рачок Р. В., Мазур В. Ю. Визначення підходів до просторового аналізу даних в інформаційно-телекомунікаційній системі морської охорони «Гарт-12» // Збірник наукових праць № 1 (75). Серія: Військові та технічні науки. – Хмельницький: Вид. НАДПСУ, 2018. – С. 134-143.

10. Боровик О. В., Рачок Р. В., Мазур В. Ю. Методика виявлення просторових аномалій руху суден та її використання при оцінюванні ризиків у системі морської охорони «Гарт-12» // Наука і оборона. – 2018. - № 2. С. 65-69.

11. Мазур В. Ю., Боровик О. В., Рачок Р. В. Метод кластеризації маршрутів суден в системі висвітлення надводної обстановки // Сучасні інформаційні технології у сфері безпеки та оборони Вип. № 2(32). – К.: Національний університет оборони України імені Івана Черняховського, Інститут інформаційних технологій, 2018. – С. 87-92.

12. Рачок Р. В., Боровик О. В., Мазур В. Ю. Методика виявлення часових аномалій руху суден на основі даних інформаційно-телекомунікаційної системи морської охорони «Гарт-12» у системі висвітлення надводної обстановки. – Наука і техніка Повітряних Сил Збройних Сил України № 3 (32). – Харків: ХНУПС, 2018. – С. 128-134.

13. Мазур В. Ю., Купрієнко Д. А., Боровик О. В. Науково-методичні рекомендації органам управління прикордонного загону щодо побудови підсистеми технічного моніторингу надводної обстановки на морській ділянці відповідальності // Честь і закон № 2(65)/2018. – Харків: НАНГУ, 2018. – С. 34-40.

14. Боровик О. В., Боровик Д. О., Костельна Т. В. Щодо необхідності удосконалення методу кластеризації маршрутів суден, як процедурного модуля

автоматизованої системи обробки даних в системі висвітлення надводної обстановки // Computer Systems and Information Technology № 2 (2), 2020. – Хмельницький: РВЦ ХНУ, 2020. – С. 37-46.

15. Експертні системи [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%B5%D1%80%D1%82%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0

16. PTV Visum [Електронний ресурс] – Режим доступу: https://apluss.pro/vprovadzhennja-programnogo-zabezpechennja/programne-zabezpechennja-ptv-visum/

17. MarineTraffic [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/MarineTraffic

18. Flightradar24 [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Flightradar24

19. Боровик О. В., Рачок Р. В., Мазур В. Ю. Визначення підходів до просторового аналізу даних в інформаційно-телекомунікаційній системі морської охорони «Гарт-12» // Збірник наукових праць № 1 (75). Серія: Військові та технічні науки. – Хмельницький: Вид. НАДПСУ, 2018. – С. 134-143.

20. Мазур В. Ю., Боровик О. В. Методичні основи оцінки ефективності функціонування єдиної системи висвітлення надводної обстановки на морській ділянці. – Наука і техніка Повітряних Сил Збройних Сил України № 2 (31). – Харків: ХНУПС, 2018. – С. 182-189.

21. Мазур В. Ю., Боровик О. В., Рачок Р. В. Метод кластеризації маршрутів суден в системі висвітлення надводної обстановки // Сучасні інформаційні технології у сфері безпеки та оборони Вип. № 2(32). – К.: Національний університет оборони України імені Івана Черняховського, Інститут інформаційних технологій, 2018. – С. 87-92.

22. Pan Sheng, Jingbo Yin Extracting Shipping Route Patterns by Trajectory Clustering Model Based on Automatic Identification System Data. - Sustainability 2018, 10.

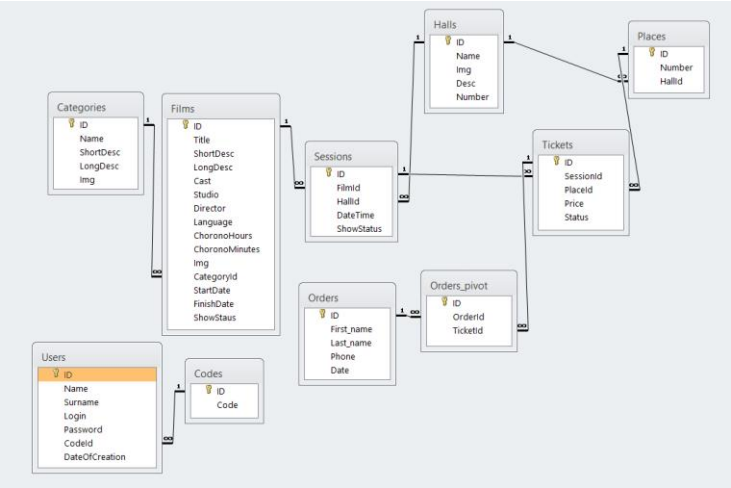
23. Боровик О. В., Рачок Р. В., Мазур В. Ю. Методика виявлення просторових аномалій руху суден та її використання при оцінюванні ризиків у системі морської охорони «Гарт-12» // Наука і оборона. – 2018. - № 2. С. 65-69.

24. Arguedas V. F., Pallotta G., Vespe M. Maritime Traffic Networks: From Historical Positioning Data to Unsupervised Maritime Traffic Monitoring. - Transactions on intelligent transportation systems, Vol. 19, № 3, March 2018. - P. 722-732.

ДОДАТКИ

Додаток А

Структура бази даних автоматизованої інформаційної системи



Додаток Б

Програмні коди

Лістинг файлу HomeController.cs

```
using CinemaTR3.Models;
using System.Linq;
using System.Web.Mvc;

namespace CinemaTR3.Controllers
{
    public class HomeController : Controller
    {
        Context db = new Context();
        public ActionResult Index()
        {
            var Users = db.Users;
            return View(Users.ToList());
        }
    }
}
```

Лістинг файлу AuthController.cs

```
using CinemaTR3.Models;
using System;
using System.Linq;
using System.Web.Mvc;

namespace CinemaTR3.Controllers
{
    public class AuthController : Controller
    {
        Context db = new Context();

        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(string
login, string password)
        {
            try
            {
                var Users = db.Users.Where(x
=> x.Login == login);
                foreach (var user in Users)
                {
                    if (password ==
user.Password || password == null)
                    {
                        if (user.Type_id ==
1)
                        {
                            ViewBag.User =
user;
                            return
View("LoginGood");
                        }
                        else if
((user.Type_id == 2))
                        {
                            ViewBag.User = user;
                            return
View("LoginGood1");
                        }
                    }
                    else
                    {
                        ViewBag.Message = "Не
вірний логін чи пароль";
                        return
View("LoginErr");
                    }
                }
            }
            catch (Exception ex)
            {
                ViewBag.Message = "Неможливо
зв'язатисьз базою даних, спробуйте пізніше";
                return View("LoginErr");
            }
            ViewBag.Message = "Заповніть усі
поля форми";
            return View("LoginErr");
        }

        [HttpGet]
        public ActionResult Home()
        {
            return View();
        }

        [HttpGet]
        public ActionResult Home1()
        {
            return View();
        }
    }
}
```

Лістинг файлу BestRouteController.cs

```
using CinemaTR3.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Data.Entity;
namespace CinemaTR3.Controllers
{
    public class BestRouteController :
Controller
    {
        Context db = new Context();

        [HttpGet]
```

```
        public ActionResult Index()
        {
            var Routes = db.BestRoutes;
            ViewBag.Routes = Routes;
            var shipTypes = db.ShipTypes;
            ViewBag.shipTypes = shipTypes;
            return View();
        }

        [HttpGet]
        public ActionResult Show(int id)
        {
            var BestRoute =
db.BestRoutes.Include(p =>
p.WaterArea).Include(p => p.ShipType).Where(x
=> x.Id == id).First();
            string[] points =
BestRoute.Points.Split(' ');
            string[] coord =
BestRoute.WaterArea.Coordinates.Split(' ');
            string Xes = "";
            string Yes = "";
            for (int i = 0; i <
points.Length; i++)
            {
                string[] pairs =
points[i].Split(';');
                Xes += pairs[0] + " ";
                Yes += pairs[1] + " ";
            }
            ViewBag.X =
Convert.ToDouble(coord[0]);
            ViewBag.Y =
Convert.ToDouble(coord[1]);

            ViewBag.Xes = Xes;
            ViewBag.Yes = Yes;
            return View(BestRoute);
        }

        [HttpGet]
        public ActionResult Create()
        {
            var WaterAreas =
db.WaterAreas.Where(x => x.IsDisplay == 1);
            ViewBag.WaterAreas = WaterAreas;
            var ShipTypes = db.ShipTypes;
            ViewBag.ShipTypes = ShipTypes;
            ViewBag.User = db.Users.Where(x
=> x.Id == 6).First();
            return View();
        }
        public ActionResult Create(BestRoute
bestRoute)
        {
            if (bestRoute.Name == null ||
                bestRoute.Points == null)
            {
                ViewBag.Message = "Не всі
поля форми заповнено коректно";
                var WaterAreas =
db.WaterAreas.Where(x => x.IsDisplay == 1);
                ViewBag.WaterAreas =
WaterAreas;
                return View();
            }
            else
            {
                //waterArea.WaterAreaType =
db.WaterAreaTypes.Where(x => x.Id ==
waterArea.Type_id).First();
                db.BestRoutes.Add(bestRoute);
                db.SaveChanges();
            }
            return Redirect("Index");
        }

        [HttpGet]
        public ActionResult Filter(int id)
        {
            var Routes =
db.BestRoutes.Where(x => x.ShipTypeId == id);
            ViewBag.Routes = Routes;
            var shipTypes = db.ShipTypes;
            ViewBag.shipTypes = shipTypes;
            return View("Index");
        }
    }
}
```

Лістинг файлу WaterAreaController.cs

```
using CinemaTR3.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Data.Entity;

namespace CinemaTR3.Controllers
{
    public class WaterAreaController :
Controller
    {
        Context db = new Context();

        [HttpGet]
        public ActionResult Index()
        {
            var WaterAreas =
db.WaterAreas.Where(x => x.IsDisplay == 1);
            return View(WaterAreas.ToList());
        }

        [HttpGet]
        public ActionResult Show(int id)
        {
            var WaterArea =
db.WaterAreas.Include(p => p.Type).Where(x =>
x.Id == id).First();
            ViewBag.WaterArea = WaterArea;
            string[] coord =
WaterArea.Coordinates.Split(' ');
```

```
            ViewBag.X =
Convert.ToDouble(coord[0]);
            ViewBag.Y =
Convert.ToDouble(coord[1]);
            return View(WaterArea);
        }

        [HttpGet]
        public ActionResult Create()
        {
            var types = db.WaterAreaTypes;
            ViewBag.types = types;
            return View();
        }
        [HttpPost]
        public ActionResult Create(WaterArea
waterArea)
        {
            if (waterArea.Name == null ||
                waterArea.Coordinates == null
||
                waterArea.Countries == null
||
                waterArea.Description ==
null)
            {
                ViewBag.Message = "Не всі
поля форми заповнено коректно";
                var types =
db.WaterAreaTypes;
                ViewBag.types = types;
                return View();
            }
            else
            {
                //waterArea.WaterAreaType =
db.WaterAreaTypes.Where(x => x.Id ==
waterArea.Type_id).First();
                db.WaterAreas.Add(waterArea);
                db.SaveChanges();
            }
            return Redirect("Index");
        }
        [HttpGet]
        public ActionResult Update(int id)
        {
            var waterArea =
db.WaterAreas.Where(x => x.Id == id).First();
            var types = db.WaterAreaTypes;
            ViewBag.types = types;
            ViewBag.waterArea = waterArea;
            return View();
        }
        [HttpPost]
        public ActionResult Update(WaterArea
waterArea)
        {
            var area = db.WaterAreas.Where(x
=> x.Id == waterArea.Id).First();
            if (waterArea.Name == null ||
                waterArea.Coordinates == null
||
                waterArea.Countries ==
null ||
                waterArea.Description ==
null)
            {
                return Redirect("../Update/"
+ area.Id);
            }
            else
            {
                area.Coordinates =
waterArea.Coordinates;
                area.Name = waterArea.Name;
                area.Countries =
waterArea.Countries;
                area.Description =
waterArea.Description;
                area.TypeId =
waterArea.TypeId;
                db.SaveChanges();
            }
            return Redirect("../Index");
        }
        public ActionResult Delete(int id)
        {
            var waterArea =
db.WaterAreas.Where(x => x.Id == id).First();
            waterArea.IsDisplay = 0;
            db.SaveChanges();
            return
Redirect("../../WaterArea/Index");
        }
    }
}
```

Лістинг файлу ObservationController.cs

```
using CinemaTR3.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Data.Entity;
namespace CinemaTR3.Controllers
{
    public class ObservationController :
Controller
    {
        Context db = new Context();

        [HttpGet]
        public ActionResult Index()
        {
            var Observations =
db.Observations.Where(x => x.IsDisplay ==
1).Where(x => x.Status == 0);
            ViewBag.Observations =
Observations;
            return View();
        }

        [HttpGet]
        public ActionResult Investigation()
```

```
        {
            var Observations =
db.Observations.Where(x => x.IsDisplay ==
1).Where(x => x.Status == 1);
            ViewBag.Observations =
Observations;
            return View();
        }

        [HttpGet]
        public ActionResult SaveAsGood(int
id)
        {
            var Observation =
db.Observations.Where(x => x.Id ==
id).First();
            Observation.Conclusion = "0.5";
            Observation.Status = 0;
            db.SaveChanges();
            return
Redirect("../Investigation");
        }

        [HttpGet]
        public ActionResult SaveAsBad(int id)
        {
            var Observation =
db.Observations.Where(x => x.Id ==
id).First();
            Observation.Conclusion = "3.5";
            Observation.Status = 0;
            db.SaveChanges();
            return
Redirect("../Investigation");
        }

        [HttpGet]
        public ActionResult
InvestigationShow(int id)
        {
            var Observation =
db.Observations.Include(p =>
p.BestRoute).Include(p =>
p.Dependency).Include(p =>
p.BestRoute.WaterArea).Include(p =>
p.BestRoute.ShipType).Where(x => x.Id ==
id).First();
            string[] points =
Observation.BestRoute.Points.Split(' ');
            string[] pointsObs =
Observation.Points.Split(' ');
            string[] coord =
Observation.BestRoute.WaterArea.Coordinates.S
plit(' ');
            string Xes = "";
            string Yes = "";
            string XesObs = "";
            string YesObs = "";
            for (int i = 0; i <
points.Length; i++)
            {
                string[] pairs =
points[i].Split(';');
                Xes += pairs[0] + " ";
                Yes += pairs[1] + " ";
            }

            for (int i = 0; i <
pointsObs.Length; i++)
            {
                string[] pairs =
pointsObs[i].Split(';');
                XesObs += pairs[0] + " ";
                YesObs += pairs[1] + " ";
            }

            ViewBag.X =
Convert.ToInt32(coord[0]);
            ViewBag.Y =
Convert.ToInt32(coord[1]);

            ViewBag.Xes = Xes;
            ViewBag.Yes = Yes;
            ViewBag.XesObs = XesObs;
            ViewBag.YesObs = YesObs;
            ViewBag.XesObs1 = XesObs;
            ViewBag.YesObs1 = YesObs;
            ViewBag.Length =
pointsObs.Length;
            ViewBag.routeId =
Observation.BestRouteId;
            ViewBag.userId = 6;

            var Dependensies =
db.Dependencies;
            ViewBag.Dependensies =
Dependensies;
            return View(Observation);
        }

        [HttpGet]
        public ActionResult Show(int id)
        {
            var Observation =
db.Observations.Include(p =>
p.BestRoute).Include(p =>
p.Dependency).Include(p =>
p.BestRoute.WaterArea).Include(p =>
p.BestRoute.ShipType).Where(x => x.Id ==
id).First();
            string[] coord =
Observation.BestRoute.WaterArea.Coordinates.S
plit(' ');

            ViewBag.X =
Convert.ToInt32(coord[0]);
            ViewBag.Y =
Convert.ToInt32(coord[1]);

            return View(Observation);
        }

        public ActionResult ToBestRoute(int
id)
        {
```

```
            var Observation =
db.Observations.Include(p =>
p.BestRoute).Include(p =>
p.Dependency).Include(p =>
p.BestRoute.WaterArea).Include(p =>
p.BestRoute.ShipType).Where(x => x.Id ==
id).First();
            Observation.IsDisplay = 0;

            var BestRoute = new BestRoute();
            BestRoute.Points =
Observation.Points;
            BestRoute.ShipTypeId =
Observation.BestRoute.ShipTypeId;
            BestRoute.WaterAreaId =
Observation.BestRoute.WaterAreaId;
            BestRoute.UserId =
Observation.UserId;
            BestRoute.Name =
Observation.Name;
            BestRoute.Created_at =
Observation.Created_at;
            db.BestRoutes.Add(BestRoute);
            db.SaveChanges();
            return Redirect("../");
        }

        [HttpGet]
        public ActionResult Select()
        {
            var WaterAreas =
db.WaterAreas.Where(x => x.IsDisplay == 1);
            ViewBag.WaterAreas = WaterAreas;
            var ShipTypes = db.ShipTypes;
            ViewBag.ShipTypes = ShipTypes;
            return View();
        }
        [HttpPost]
        public ActionResult
Select(TransferObject obj)
        {
            var waterAreaId = obj.AreaId;
            var shipTypeId = obj.ShipId;
            var Routes =
db.BestRoutes.Where(x => x.ShipTypeId ==
shipTypeId).Where(x => x.WaterAreaId ==
waterAreaId);
            var waterArea =
db.WaterAreas.Where(x => x.Id ==
waterAreaId).First();
            var shipType =
db.ShipTypes.Where(x => x.Id ==
shipTypeId).First();
            ViewBag.Routes = Routes;
            ViewBag.waterArea = waterArea;
            ViewBag.shipType = shipType;
            return View("RouteList");
        }
        [HttpGet]
        public ActionResult Create(int id)
        {
            var Route =
db.BestRoutes.Include(p =>
p.WaterArea).Include(p =>
p.ShipType).Where(x => x.Id == id).First();
            string[] points =
Route.Points.Split(' ');
            string[] coord =
Route.WaterArea.Coordinates.Split(' ');
            string Xes = "";
            string Yes = "";
            for (int i = 0; i <
points.Length; i++)
            {
                string[] pairs =
points[i].Split(';');
                Xes += pairs[0] + " ";
                Yes += pairs[1] + " ";
            }
            ViewBag.X =
Convert.ToDouble(coord[0]);
            ViewBag.Y =
Convert.ToDouble(coord[1]);

            ViewBag.Xes = Xes;
            ViewBag.Yes = Yes;
            ViewBag.Length = points.Length;
            ViewBag.routeId = id;
            ViewBag.userId = 6;

            var Dependensies =
db.Dependencies;
            ViewBag.Dependensies =
Dependensies;
            return View(Route);
        }
        public ActionResult
Create(Observation obs)
        {
            obs.DependencyId =
Convert.ToInt32(obs.DependencyId);
            obs.UserId =
Convert.ToInt32(obs.UserId);
            obs.BestRouteId =
Convert.ToInt32(obs.BestRouteId);
            obs.Points =
obs.Points.Remove(obs.Points.Length - 1, 1);
            obs.IsDisplay = 1;
            db.Observations.Add(obs);
            db.SaveChanges();

            return
Redirect("../../Auth/Home");
        }

        /*[HttpGet]
        public ActionResult Filter(int id)
        {
            var Routes =
db.BestRoutes.Where(x => x.ShipTypeId == id);
            ViewBag.Routes = Routes;
            var shipTypes = db.ShipTypes;
            ViewBag.shipTypes = shipTypes;
            return View("Index");
        }*/
    }
```

```
}
```

Лістинг файлу BestRoute.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace CinemaTR3.Models
{
    public class BestRoute
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public int? WaterAreaId { get; set; }
        public WaterArea WaterArea { get;
set; }
        public int? UserId { get; set; }
        public User User { get; set; }
        public string Points { get; set; }
        public DateTime Created_at { get;
set; }
        public int? ShipTypeId { get; set; }
        public ShipType ShipType { get; set;
}
    }
}
```

Лістинг файлу WaterArea.cs

```
using System;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
namespace CinemaTR3.Models
{
    public class WaterArea
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Countries { get; set; }
        public string Coordinates { get; set;
}
        public string Description { get; set;
}
        public int? TypeId { get; set; }
        public WaterAreaType Type { get; set;
}
        public DateTime Created_at { get;
set; }
        public int? IsDisplay { get; set; }
    }
}
```

Лістинг файлу WaterAreaType.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace CinemaTR3.Models
{
    public class WaterAreaType
    {
        public int Id { get; set; }
        public string Name { get; set; }

    }
}
```

Лістинг файлу Context.cs

```
using System;
using System.Data.Entity;

namespace CinemaTR3.Models
{
    public class Context : DbContext
    {

        public DbSet<User> Users { get; set;
}
        public DbSet<WaterArea> WaterAreas {
get; set; }
        public DbSet<UserType> UserTypes {
get; set; }
        public DbSet<WaterAreaType>
WaterAreaTypes { get; set; }
        public DbSet<ShipType> ShipTypes {
get; set; }
        public DbSet<Dependency> Dependencies
{ get; set; }
        public DbSet<BestRoute> BestRoutes {
get; set; }
        public DbSet<Observation>
Observations { get; set; }

        public UserType UserType
        {
            get => default;
            set
            {
            }
        }

        public ShipType ShipType
        {
            get => default;
            set
            {
            }
        }

        public WaterAreaType WaterAreaType
        {
            get => default;
            set
            {
            }
        }

        public BestRoute BestRoute
        {
            get => default;
            set
            {
            }
```

```
        }

        public Observation Observation
        {
            get => default;
            set
            {
            }
        }

        public TransferObject TransferObject
        {
            get => default;
            set
            {
            }
        }

        public Dependency Dependency
        {
            get => default;
            set
            {
            }
        }

        public WaterArea WaterArea
        {
            get => default;
            set
            {
            }
        }

        public User User
        {
            get => default;
            set
            {
            }
        }
    }
}
```

Лістинг файлу Dependency.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace CinemaTR3.Models
{
    public class Dependency
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }
}
```

Лістинг файлу Observation.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace CinemaTR3.Models
{
    public class Observation
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public int? BestRouteId { get; set; }
        public BestRoute BestRoute { get;
set; }
        public int? UserId { get; set; }
        public User User { get; set; }
        public int? DependencyId { get; set;
}
        public Dependency Dependency { get;
set; }
        public string Points { get; set; }
        public string Conclusion { get; set;
}
        public DateTime Created_at { get;
set; }
        public int IsDisplay { get; set; }
        public int Status { get; set; }
    }
}
```

Лістинг файлу User.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace CinemaTR3.Models
{
    public class User
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Surname { get; set; }
        public string Login { get; set; }
        public string Password { get; set; }
        public int? Type_id { get; set; }
        public UserType Type { get; set; }
        public DateTime DateOfCreation { get;
set; }
    }
}
```

Лістинг файлу UserType.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace CinemaTR3.Models
{
    public class UserType
    {
        public int Id { get; set; }
        public string Name { get; set; }

    }
```

```
}
```

Лістинг файлу ObservationCreate.cs

```
@model CinemaTR3.Models.BestRoute

@{
    ViewBag.Title = "Create";
}

<div    style="text-align:    center;    margin-
top:30px;">
    <h2>New observation</h2>
    <p>Water  area:   @Model.WaterArea.Name  and
ship type: @Model.ShipType.Name</p>
</div>
<div   class="list-wrap"   style="   background:
white; min-height: 86vh; padding: 30px;">
    <h2   style="text-align:   center;   margin-
bottom: 20px;">@Model.Name</h2>
    <p     class="status"     style="background:
#e3fdd8; color: green; display: none; text-align:
center; ">Satisfactory route </p>
    <p     class="status"     style="background:
#fdfad8; display: none; text-align: center;">Pay
attention to this route</p>
    <p     class="status"     style="background:
#fdd8d8; color: red; display: none; text-align:
center;">Danger. Reaction needed</p>
    <div style="width:100%;">
        <div id="graph"></div>
    </div>
    <div class="area-wrap">
        <div id="map1" style="width:50%;"></div>
        <div id="info">
            <p><strong>Input points</strong></p>
            <form method="post">
                <div class="form-group">
                    <label for="Dep">Choose math
dependensie</label>
                    <select  name="DependencyId"
class="form-control depSelect" id="Dep">
                        @foreach   (var   p   in
ViewBag.Dependensies)
                        {
                            <option
value="@p.Id">@p.Name</option>
                        }
                    </select>
                </div>
                <div class="form-group">
                    @for  (var  i  =  0;  i  <
ViewBag.Length; i++)
                    {
                        <div
style="display:flex;">
                            <div>
                                <label
for="X">X:</label>
                                <input
type="number"  name="x"  class="x-control"  id="X"
/>
                            </div>
                            <div>
                                <label
for="Y">Y:</label>
                                <input
type="number"  name="y"  class="y-control"  id="Y"
/>
                            </div>
                            <div class="done">
                                Save
                            </div>
                        </div>
                    }
                </div>
                <div class="form-group">
                    <input      type="checkbox"
id="scales" name="Status"
                           checked>
                    <label
for="scales">Additional            investigation
needed</label>
                </div>
                <input type="hidden" name="Name"
value="@Model.WaterArea.Name
@Model.ShipType.Name @DateTime.Now" />
                <input             type="hidden"
name="BestRouteId" value="@ViewBag.routeId" />
                <input             type="hidden"
name="UserId" value="@ViewBag.userId" />
                <input      class="route-points"
type="hidden" name="Points" value="" />
                <input              class="conc"
type="hidden" name="Conclusion" value="" />
                <input             type="hidden"
name="Created_at" value="@DateTime.Now" />
                <input  type="submit"  disabled
name="sub" class="sub" value="Save observation"
/>
            </form>
        </div>
    </div>

</div>
<script        src="https://cdn.plot.ly/plotly-
1.35.2.min.js"></script>
<script
src="https://maps.googleapis.com/maps/api/js?ke
y=YOUR_API_KEY&callback=initMap1&libraries=&v=w
eekly" async></script>
<script>
    let Result = 0;
    const            selectElement            =
document.querySelector('.depSelect');
    let DEP = 1;
    selectElement.addEventListener('change',
(event) => {
        DEP = event.target.value
        if (DEP == 1) {
            buildMainLinear()
            buildObsLinear()
            SplineCalculation()
            Result = Result-0.2
            changeView()
        }
        if (DEP == 2) {
            buildMainSpline()
            buildObsSpline()
            SplineCalculation()
            changeView()
        }
```

```
    });
    let Xes = '@ViewBag.Xes';
    let Yes = '@ViewBag.Yes';
    const xValues = Xes.split(" ")
    const yValues = Yes.split(" ")
    yValues.pop()
    xValues.pop()
    console.log(xValues, yValues)
    var xx1 = []
    var yy1 = []
    var xx = []
    var yy = []
    let calcX = [];
    let calcY = [];
    let YRES = [];
    let calculate = [];
    let pointRes = [];

    let  body  =  document.querySelector('.list-
wrap');

    let splines = [];
    function BuildSpline(x, y, n) {
        splines = []
        for (var i = 0; i < n; ++i) {
            splines.push({ x: 0, a: 0 });
            splines[i].x = x[i];
            splines[i].a = y[i];
        }
        splines[0].c = splines[n - 1].c = 0.0;

        let alpha = [];
        let beta = [];
        alpha[0] = beta[0] = 0.0;
        for (var i = 1; i < n - 1; ++i) {
            var hi = x[i] - x[i - 1];
            var hi1 = x[i + 1] - x[i];
            var A = hi;
            var C = 2.0 * (hi + hi1);
            var B = hi1;
            var F = 6.0 * ((y[i + 1] - y[i]) /
hi1 - (y[i] - y[i - 1]) / hi);
            var z = (A * alpha[i - 1] + C);
            alpha[i] = -B / z;
            beta[i] = (F - A * beta[i - 1]) / z;
        }

        for (var i = n - 2; i > 0; --i) {
            splines[i].c = alpha[i] * splines[i
+ 1].c + beta[i];
        }

        for (var i = n - 1; i > 0; --i) {
            var hi = x[i] - x[i - 1];
            splines[i].d   =   (splines[i].c   -
splines[i - 1].c) / hi;
            splines[i].b   =   hi   *   (2.0   *
splines[i].c + splines[i - 1].c) / 6.0 + (y[i] -
y[i - 1]) / hi;
        }
    }

    function Interpolate(x) {
        if (splines == null) {
            return NaN;
        }
        var k = 0
        let n = splines.length;
        let s = {};

        if (x <= splines[0].x) {
            s = splines[0];
        }
        else if (x >= splines[n - 1].x) {
            s = splines[n - 1];
        }
        else {
            var i = 0;
            if (n % 2== 0) {

                var j = n;
            }
            else {
                var j = n - 1;
            }
            while (i + 1 < j) {

                k = i + (j - i) / 2;
                if (x <= splines[k].x) {
                    j = k;
                }
                else {
                    i = k;
                }
            }
            s = splines[j];
        }

        let dx = x - s.x;
        if (!s.b) {
            s.b=0
        }
        return s.a + (s.b + (s.c / 2.0 + s.d * dx
/ 6.0) * dx) * dx;
    }
    function buildClaster() {
        for (var i = 0; i < xValues.length; i++)
{
            xValues[i] = Number(xValues[i])
            yValues[i] = Number(yValues[i])
        }
    }
    function buildMainLinear() {
        for (var i = 0; i < xValues.length; i++)
{
                    xValues[i]                =
Number(xValues[i])
                    yValues[i]                =
Number(yValues[i])
        }
        draw(xValues, yValues, [], [])
    }
    function buildObsLinear() {

        for (var i = 0; i < xValues.length; i++)
{
            xValues[i] = Number(xValues[i])
            yValues[i] = Number(yValues[i])
        }

        draw(xValues, yValues, XRES, YRES)
    }
```

```
        function buildMainSpline() {
            xx1 = []
            yy1 = []
                for (var i = 0; i < xValues.length; i++) {
                    xValues[i] = Number(xValues[i])
                    yValues[i] = Number(yValues[i])
                }
                if (xValues.length > 1) {
                    BuildSpline(xValues, yValues, xValues.length)
                    for (var i = xValues[0]; i <= xValues[xValues.length - 1]; i = i + 0.25) {
                        xx1.push(i)

yy1.push(Interpolate(i))
                    }

                    yy1[0] = yValues[0]
                    draw(xx1, yy1, [], [])
            }


    }
    function buildObsSpline() {

        if (XRES.length > 1) {
            xx = []
            yy = []
            BuildSpline(XRES, YRES, XRES.length)


            for (var i = XRES[0]; i <= XRES[XRES.length - 1]; i = i + 0.25) {
                xx.push(i)
                yy.push(Interpolate(i))
            }
            yy[0] = YRES[0]
            draw(xx1, yy1, xx, yy)
        }

    }


    let xesInputs = document.querySelectorAll('.x-control');
    let yesInputs = document.querySelectorAll('.y-control');
    let dones = document.querySelectorAll('.done');
    let XES = [];
    let YES = [];
    let XRES = [];
    let YRES1 = [];


    for (var i = 0; i < dones.length; i++) {
        dones[i].addEventListener("click", (event) => {
            var elements = [...dones]

            let index = 0;

            for (var j = 0; j < elements.length; j++) {
                if (elements[j] == event.target) {
                    index = j;
                }
            }

            if (xesInputs[index].value !== '' && yesInputs[index].value !== '') {
                xesInputs[index].disabled = true;
                yesInputs[index].disabled = true;

XES.push(Number(xesInputs[index].value));
                YES.push({ value: yesInputs[index].value, position: index });
                XRES = XES.sort(compareNumbers);
                YRES1 = YES.sort(comparePositions);
            }
            YRES = []
            for (var i = 0; i < YRES1.length;i++) {

YRES.push(Number(YRES1[i].value));
            }
            if (DEP == 1) {

                buildMainLinear()
                buildObsLinear()
                SplineCalculation()
                changeView()
            }
            if (DEP == 2) {
                buildMainSpline()
                buildObsSpline()
                SplineCalculation()
                changeView()
            }
        });
    }
            function compareNumbers(a, b) {
                return a - b;
            }
            function comparePositions(a, b) {
                return a.position - b.position;
            }

    function draw(Xes, Yes, Xres, Yres) {
        try {
            const trace1 = {
                x: Xes,
                y: Yes,
                type: 'scatter',
                name: 'Graph of route'
            }
            const trace2 = {
                x: Xres,
                y: Yres,
                type: 'scatter',
                name: 'Graph of observation route'
            }
            const data = [trace1, trace2]
```

```
            Plotly.newPlot('graph', data)
        }
        catch (err) {
            console.error(err)
            alert(err)
        }
    }

    function SplineCalculation() {
        for (var i = 0; i < xValues.length; i++)
{
            xValues[i] = Number(xValues[i])
            yValues[i] = Number(yValues[i])
        }
        if (xValues.length > 1) {
            BuildSpline(xValues,        yValues,
xValues.length)
            if (calcX.length == 0) {
                for (var i = xValues[0]; i <=
xValues[xValues.length - 1]; i = i + 0.1) {
                    calcX.push(i)
                    calcY.push(Interpolate(i))
                }
                calcY[0] = yValues[0]
            }

            if (XRES.length > 1) {
                for (let j = 0; j < XRES.length;
j++) {
                    for (let k = 0; k <
calcX.length; k++) {

pointRes.push(twoPointDistance(XRES[j], YRES[j],
calcX[k], calcY[k]))
                    }

calculate.push(Math.min(...pointRes))
                    pointRes = []
                }
                var temp = 0
                for (let i = 0; i <
calculate.length; i++) {
                    temp += calculate[i]
                }
                console.log(calculate)
                Result      =      temp      /
calculate.length;
                calculate = []
            }
        }
    }

    let                statuses                =
document.querySelectorAll(".status")
    function changeView() {
        if (XRES.length > 1) {
            for (let i = 0; i < statuses.length;
i++) {
                statuses[i].style.display      =
"none";
            }
            if (Result < 1.5) {
                statuses[0].style.display      =
"block";
                body.style.background      =
"#e3fdd8";
            }
            else if (Result >= 1.5 && Result <=
2.5) {
                statuses[1].style.display      =
"block";
                body.style.background          =
"#fdfad8";
            }
            else {
                statuses[2].style.display      =
"block";
                body.style.background          =
"#fdd8d8";
            }
        }
        let               points               =
document.querySelector(".route-points")
        let                conc                =
document.querySelector(".conc")
        let sub = document.querySelector(".sub")
        sub.disabled = false
        conc.value = Result
        let str = ""
        for (let i = 0; i < XRES.length; i++) {
            str += XRES[i] + ';' + YRES[i] + ' ';
        }
        points.value = str
    }

    function twoPointDistance(x, y, x1, y1) {
        return Math.sqrt(Math.pow((x - x1),2) +
Math.pow((y - y1),2))
    }

    function initMap1() {
        const uluru = { lat: @ViewBag.X, lng:
@ViewBag.Y };
        const         map         =         new
google.maps.Map(document.getElementById("map1")
, {
            zoom: 4,
            center: uluru,
        });
        const marker = new google.maps.Marker({
            position: uluru,
            map: map,
        });
    }
    window.addEventListener("load",     function
(event) {
       buildMainLinear()
    });
</script>
```

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

МЕТОД ОЦІНКИ ВІДПОВІДНОСТІ МАРШРУТІВ КОРАБЛІВ ДО ЕТАЛОННОГО МАРШРУТУ

Виконав:
студент 4 курсу, групи КН-18-1
Боровик Дмитро Олегович

Керівник:
д.т.н., професор, завідувач кафедри КН
Бармак Олександр Володимирович

Актуальність

Інформаційна складова сучасної моделі охорони кордону на морській ділянці реалізується з використанням інтегрованої інформаційно-телекомунікаційної системи Морської охорони «Гарт-12». Застосування цієї системи дозволяє отримувати інформацію про поточне положення суден, порт відправки та порт призначення, тип вантажу та інші додаткові дані. При виборі судна у системі висвітлення надводної обстановки існує можливість перегляду попередніх точок його маршруту, отриманих у певні дискретні моменти часу. Останнє необхідне для встановлення можливих ознак порушення правил прикордонного режиму порушниками прикордонного законодавства. При цьому, візуалізація маршруту здійснюється з використанням лінійної апроксимації, а можливе порушення правил прикордонного режиму встановлюється на основі порівняння маршруту руху окремого судна та еталонного маршруту з кластера маршрутів, що з'єднують пункт відправлення та призначення судна. Однак, застосування лінійної апроксимації при формуванні неперервного апроксимаційного маршруту руху судна є сумнівним і нічим не підтвердженим.

Тому актуальним є дослідження впливу виду апроксимації при побудові в системі висвітлення надводної обстановки неперервного маршруту руху судна через множину дискретних точок місцезнаходження судна у фіксовані моменти часу на величину метрики, що застосовується для встановлення подібності маршруту двох різних суден.

Мета і задачі роботи

Метою кваліфікаційної роботи бакалавра є створення методу порівняння маршрутів кораблів з еталонним маршрутом та розробка інформаційної системи реалізації зазначеного методу.

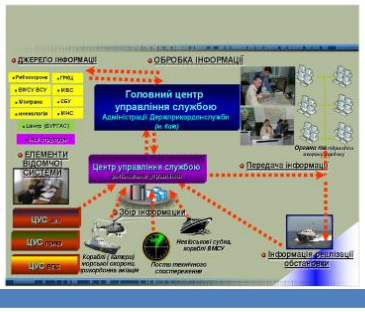
Часткові задачі дослідження:

- *аналіз предметної області та її інформаційного забезпечення;*
- *обґрунтування методу порівняння відповідності маршрутів суден еталонному маршруту;*
- *проектування структури інформаційної системи реалізації методу порівняння відповідності маршрутів суден еталонному маршруту;*
- *програмна реалізація методу порівняння відповідності маршрутів суден еталонному маршруту.*

Об'єктом дослідження є маршрутизація руху суден в системі висвітлення надводної обстановки.

Предметом дослідження є методи встановлення відповідності між маршрутами руху суден і еталонним маршрутом в системі висвітлення надводної обстановки.

Система висвітлення надводної обстановки ДПСУ



Існуюче програмне забезпечення предметної області

Інтерфейс експертної системи «PTV Visum»

Інтерфейс системи «MarineTraffic»

Інтерфейс системи «FlightRadar24»

Недоліки існуючого програмного забезпечення предметної області:

- незадовільна швидкодія;
- висока вартість;
- обмежена доступність;
- обмежена функціональна придатність.

Постановка задачі

- Обгрунтувати метод для порівняння маршрутів кораблів відносно еталонного маршруту для систем висвітлення надводної обстановки та створити відповідну інформаційну систему для оцінки ступеня відхилення маршруту судна від еталонного маршруту і прийняття рішення щодо необхідності реагування на ймовірне порушення правил прикордонного законодавства.
- Застосунок має використовуватись суб'єктами забезпечення прикордонної безпеки для підвищення ефективності виявлення порушників прикордонного законодавства.
- Розробку web-застосунку доцільно здійснити на платформі .NET Framework з використанням мови програмування C# та фреймворку даної мови – ASP .Net.
- Відповідна інформаційна система призначена для автоматизації роботи таких користувачів, як оператор ЦУС регіонального управління Морської охорони, оператор Головного ЦУС Адміністрації Держприкордонслужби.

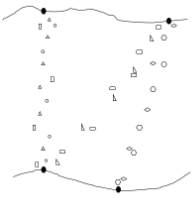
Комплекс взаємопов'язаних задач, вирішення яких забезпечує можливість встановлення порушення судном правил прикордонного режиму

- встановлення подібності маршрутів довільних пар різних суден;
- встановлення кластеру маршрутів між заданими пунктами відправлення та призначення (транспортного коридору між двома пунктами);
- визначення еталонного маршруту між двома пунктами в межах встановленого кластеру;
- встановлення належності траєкторії руху судна кластеру маршрутів між заданими пунктами відправлення та призначення;
- оцінка ступеню відхилення траєкторії досліджуваного судна від еталонного маршруту.

Обгрунтування методу порівняння маршрутів кораблів

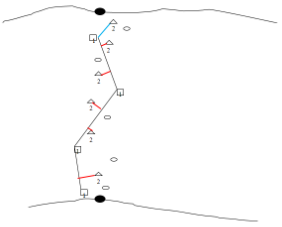
У базі даних ІТС морської охорони «Гарт-12» інформація про пересування суден зберігається у вигляді впорядкованої послідовності точок з координатами суден у певні моменти часу.

Уявлення про місце знаходження суден у фіксований момент часу можна сформувати з використанням рисунку.



Схематичне представлення даних щодо положення окремих суден у СВНО

Встановлення подібності маршрутів довільних пар різних суден



Підхід до визначення ступеня подібності маршрутів за допомогою метрики

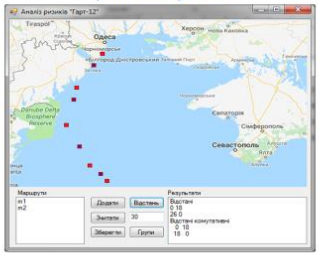
Ступінь подібності маршрутів M_1 та M_2 запропоновано визначати за допомогою метрики $R(M_1,M_2)$.

Для розрахунку $R(M_1,M_2)$ необхідно насамперед здійснити встановлення (відновлення, генерування) неперервного маршруту руху першого судна між заданими у дискретній множині M_1 точками з використанням лінійної апроксимації. Величина $R(M_1,M_2)$ визначається, як середнє відхилення точок множини M_2, які описують дискретний маршрут руху другого судна, від апроксимованого неперервного першого маршруту у вигляді

$$R(M_1,M_2)=\frac{\sum_{j=1}^{k_2} R_{min\,j}}{k_2},$$

де $R_{min\,j}$ є найкоротшою відстанню від j-ї точки множини M_2 до встановленого першого маршруту, тобто до найближчого деякого i-го сегменту прямої, який з'єднує точки (x_i,y_i) і (x_{i+1},y_{i+1}) множини M_1, а k_2 - кількість досліджуваних точок другого маршруту.

Встановлення подібності маршрутів довільних пар різних суден



Розрахунок метрик $R(M_1,M_2)$ і $R(M_2,M_1)$

Встановлення подібності маршрутів довільних пар різних суден

Приклади обчислення метрики

Встановлення кластеру маршрутів між заданими пунктами відправлення та призначення (транспортного коридору між двома пунктами)

Введення деякого граничного порогового рівня r_{gk} (даний рівень визначається на основі експертної оцінки) щодо величини відхилення маршрутів і його порівняння з метрикою дозволяє робити висновки про ступінь подібності маршрутів M_i і M_j, незалежно від того, для якого з маршрутів формується апроксимаційна неперервна траєкторія руху, і відносно якого здійснюється оцінка близькості до такої траєкторії.

Встановлення геометричної близькості маршрутів за допомогою метрики дає можливість провести їх подальшу класифікацію та поділ на кластери (кластеризацію) з подальшим визначенням у кожному з них еталонного маршруту. Еталонні маршрути можуть використовуватись для оцінки ступеня відхилення судна, відносно якого проводиться аналіз. Також може здійснюватись подальша класифікація нових даних, що поступають до системи, та оновлення або уточнення меж кластера.

Для проведення кластеризації маршрутів на основі наявних у базі «Гарт-12» даних пропонується побудувати матрицю $\boldsymbol{R}$, елементи r_{ij} якої визначаються наступним чином

$$r_{ij} = \begin{cases} 0, i = j, \\ R^k(M_i, M_j), i \neq j, \end{cases}$$

де M_i, M_j - маршрути відповідно i-го та j-го суден.

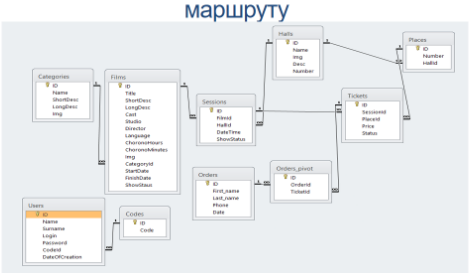
Після формування матриці $\boldsymbol{R}$ для кожного i рядка обчислюється:

- кількість n_i елементів, які не перевищують порогового для кластеризації рівня r_{gk}, тобто кількість елементів, для яких у рядку i виконується нерівність $R^k(M_i, M_j) \leq r_{gk}$;
- величина $s_i = \sum R^k(M_i, M_j)$, яка формується лише з тих значень $R^k(M_i, M_j)$, для яких виконується умова $R^k(M_i, M_j) \leq r_{gk}$.

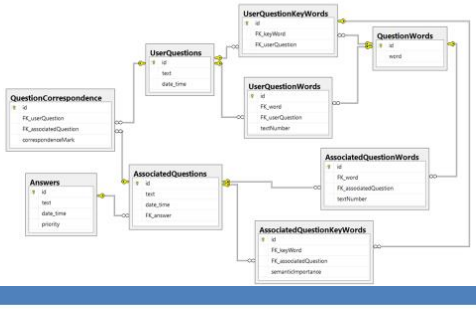
На основі отриманих даних, які сортуються за зменшенням n_i (для елементів з однаковими значеннями n_i здійснюється сортування за збільшенням s_i), формується впорядкована множина $\boldsymbol{M_k}$, елементами якої є набори $(i,\ n_i(j,\ s_i(j)))$.

Далі множина $\boldsymbol{M_k}$ використовується для визначення тих маршрутів, які можуть бути включені до одного і того ж кластера.

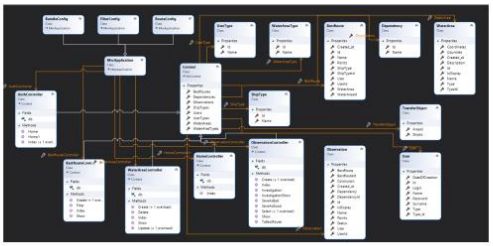
Даталогічна структура бази даних AIC оцінки ступеня відхилення маршруту судна від еталонного маршруту



Даталогічна модель бази даних



Діаграма класів програмної реалізації методу оцінки відповідності маршрутів кораблів еталонному маршруту



Діаграма активності при дослідженні маршруту

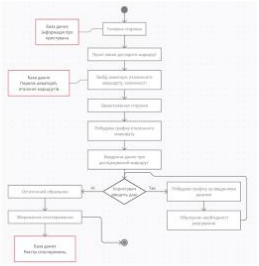
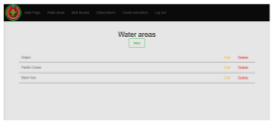
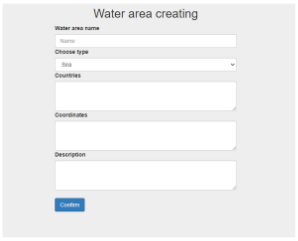


Схема роботи користувача з інформаційною системою

Програмна реалізація методу оцінки відповідності маршрутів кораблів еталонному маршруту

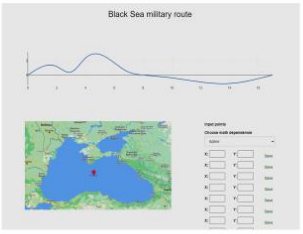


Додавання акваторії до бази даних

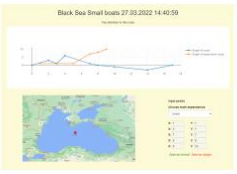


Програмна реалізація методу оцінки відповідності маршрутів кораблів еталонному маршруту

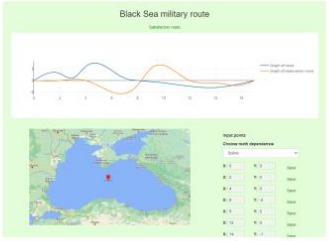
Створення спостереження



Програмна реалізація методу оцінки відповідності маршрутів кораблів еталонному маршруту



Отримання результатів дослідження маршруту



Апробація кваліфікаційної роботи

1. Боровик О. В., Боровик Д. О., Костельна Т. В. Щодо необхідності удосконалення методу кластеризації маршрутів суден, як процедурного модуля автоматизованої системи обробки даних в системі висвітлення надводної обстановки // Computer Systems and Information Technology № 2 (2), 2020. – Хмельницький: РВЦ ХНУ, 2020. – С. 37-46.
2. Боровик О. В., Боровик Д. О. Особливості встановлення величини метрики, що застосовується для ідентифікації подібності маршрутів суден в системі висвітлення надводної обстановки // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка.. Вип. № 70. – К: ВІКНУ, 2021. – С. 21-41.
3. Боровик О. В., Боровик Д. О. Шляхи удосконалення методу кластеризації маршрутів суден // Збірник тез доповідей Міжнародної науково-практичної конференції (10-11 вересня 2020 року) «Спільні дії військових формувань і правоохоронних органів держави: проблеми та перспективи». Секція №5 Використання сучасних інформаційних технологій у секторі безпеки і оборони: проблеми та розвиток. Сучасні проблеми моделювання подій, явищ і процесів. – Одеса: Друкарня Військової академії (м. Одеса), 2020. – С. 256-257.
4. Олег БОРОВИК, Дмитро БОРОВИК Вплив методу кластеризації маршрутів суден на достовірність результатів моделювання в автоматизованій системі обробки даних системи висвітлення надводної обстановки // Тези XII-І Всеукраїнської науково-практичної конференції (26 листопада 2020 року) «Освітньо-наукове забезпечення діяльності складових сектору безпеки й оборони України» – Хмельницький.: Вид. НАДПСУ, 2020. – С. 841-843.
5. Боровик О. В., Боровик Д. О. Шляхи удосконалення інформаційно-телекомунікаційної системи висвітлення надводної обстановки // Матеріали Всеукраїнської науково-практичної Інтернет-конференції (11-21 березня 2021 року) «Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку» – Черкаси: ЧНУ, 2021. – С. 107-109.
6. Бармак О. В., Боровик Д. О. Щодо актуальності розбудови інформаційної системи оцінки відповідності апроксимаційних маршрутів кораблів еталонним маршрутам // Збірник тез доповідей XVII Міжнародної науково-практичної конференції (26 листопада 2021 року) «Військова освіта і наука: сьогодення та майбутнє». Секція №5 Сучасні інформаційно-комунікаційні технології сектору безпеки і оборони України. – Том 2. - К.: ВІКНУ, 2021. – С. 93-94.

Висновки

Під час виконання кваліфікаційної роботи бакалавра був проведений аналіз предметної області, пов'язаної із національною безпекою та запобіганням контрабанді. Також було проаналізовано існуюче програмне забезпечення, що зараз використовується у держструктурах. Було поставлено завдання щодо розробки автоматизованої інформаційної системи порівняння маршрутів кораблів відносно еталонних маршрутів із веб-інтерфейсом та функціями, що необхідні для функціонування даного типу програмного забезпечення.

У результаті виконання кваліфікаційної роботи було обгрунтовано метод для порівняння маршрутів кораблів відносно еталонного маршруту у системі висвітлення надводної обстановки та створено відповідну інформаційну систему для оцінки ступеня відхилення маршруту судна від еталонного маршруту та прийняття рішення щодо необхідності реагування на ймовірне порушення правил прикордонного законодавства. Для розробки програмного продукту було використано мову програмування C#, також платформу, що є фреймворком мови C# – ASP.Net MVC та СКБД – SQL Server.

ДЯКУЮ ЗА УВАГУ!

Thu May 26 16:11:10 EEST 2022, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 2.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 12%**

	Документ		Суммарное совпадение по Базе Данных	
ID: 104053 Название: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Метод оцінки відповідності маршрутів кораблів до еталонного маршруту Добавлено в БД: 2022-05-26 Авторы: Д.О. Боровик Руководители: О.В. Бармак Консультанты: Опоненты:	Символы	Лексемы	Символы	Лексемы
	66065	961	3199 (5%)	49 (5%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

UNICHECK

Ім'я користувача:
Кафедра КН

ID перевірки:
1011344728

Дата перевірки:
26.05.2022 17:02:05 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
26.05.2022 17:08:44 EEST

ID користувача:
100005671

Назва документа: **Боровик_ЗАПИСКА_short**

Кількість сторінок: **71** Кількість слів: **10648** Кількість символів: **80175** Розмір файлу: **2.84 MB** ID файлу: **1011230523**

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

27.4%
Схожість

Найбільша схожість: **18%** з Інтернет-джерелом (http://elar.khnu.km.ua/jspui/bitstream/123456789/9960/1/23-%D0%A2%..

24.8% Джерела з Інтернету	79	Сторінка 73
17.9% Джерела з Бібліотеки	128	Сторінка 74

0.08% Цитат

Цитати	1	Сторінка 75

Не знайдено жодних посилань

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	68
Підозріле форматування	13 сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: *Метод оцінки відповідності маршрутів кораблів до еталонного маршруту*

Автор: *студент групи КН-18-1 Боровик Дмитро Олегович*

Спеціальність: *122 – Комп'ютерні науки*

Освітня програма: *освітньо-професійна*

Науковий керівник: *д.т.н., проф. Бармак О.В.*

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	***відповідає***
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі Боровика Д.О., не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції; серед запозичень знаходяться загальновідомі терміни, скорочення та матеріали статей, серед автрів яких є Боровик Д.О..

Обсяг запозичень, визначений системами виявлення збігів/ ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 2%;

- за системою Unicheck: 27.4 %, з яких 18% є посиланням на власну статтю "ЩОДО НЕОБХІДНОСТІ УДОСКОНАЛЕННЯ МЕТОДУ КЛАСТЕРИЗАЦІЇ МАРШРУТІВ СУДЕН, ЯК ПРОЦЕДУРНОГО МОДУЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ОБРОБКИ ДАНИХ В СИСТЕМІ ВИСВІТЛЕННЯ НАДВОДНОЇ ОБСТАНОВКИ" у журналі «COMPUTER SYSTEMS AND INFORMATION TECHNOLOGIES» N2, 2020. Отже 27.4-18=9.4% є допустимими запозиченнями які відносяться до описаних вище.

Керівник роботи _____ *Олександр БАРМАК*

Гарант ОП _____ *Олександр МАЗУРЕЦЬ*

Завідувач кафедри КН _____ *Олександр БАРМАК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента *гр. КН-18-1 Боровик Дмитро Олегович*
за темою: *Метод оцінки відповідності маршрутів кораблів до еталонного маршруту*

1. Актуальність обраної теми

Задача встановлення можливих ознак порушення правил прикордонного режиму порушниками прикордонного законодавства на морі є важливою з точки зору забезпечення належного рівня національної безпеки. Її вирішення пов'язується з використанням інтегрованої інформаційно-телекомунікаційної системи Морської охорони «Гарт-12». У цій системі візуалізація маршруту руху судна здійснюється на основі застосування лінійної апроксимації, а можливе порушення правил прикордонного режиму встановлюється на основі порівняння маршруту руху окремого судна та еталонного маршруту з кластера маршрутів, що з'єднують пункт відправлення та призначення судна.

Зважаючи на те, що застосування лінійної апроксимації при формуванні неперервного апроксимаційного маршруту руху судна є сумнівним і нічим не підтвердженим, актуальним питанням є дослідження впливу виду апроксимації на величину метрики, що застосовується для встановлення подібності маршруту двох різних суден. Крім цього, актуальним є і питання програмно-алгоритмічної реалізації вирішення вказаної задачі, що може сприяти автоматизації досліджуваних процесів.

2. Повнота розкриття мети та завдань роботи

Метою кваліфікаційної роботи бакалавра було опрацювання методу порівняння відповідності маршрутів суден відносно еталонного маршруту та розробка інформаційної системи реалізації вказаного методу.

Для досягнення мети роботи її автором вирішувалися наступні часткові завдання:

- *проводився аналіз предметної області та її інформаційного забезпечення;*
- *обґрунтовувався метод порівняння відповідності маршрутів суден відносно еталонного маршруту;*
- *здійснювалися проектування структури інформаційної системи та її програмна реалізація.*

Усі завдання опрацьовані достатньо повно, що дозволяє зробити висновок про повноту опрацювання теми роботи.

3. Зміст кожного розділу роботи

Наведені часткові завдання опрацьовані в трьох розділах кваліфікаційної роботи.

Перший розділ присвячений характеристиці предметної області та постановці задачі дослідження.

У другому розділі було здійснено опис методу порівняння маршрутів кораблів та проектування структури відповідної інформаційної системи.

Третій розділ присвячений програмній реалізації спроектованої у розділі 2 інформаційної системи.

4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблені автором роботи метод та інформаційна система дозволяє автоматизувати порівняння маршруту руху окремого судна та еталонного маршруту з кластера маршрутів, що з'єднують пункт відправлення та призначення судна. Її цінність

полягає в тому, що вона може бути окремим модулем оцінки достовірності висновків, що забезпечуються діючою системою висвітлення надводної обстановки.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота виконана на належному науково-методичному рівні та відповідає встановленим вимогам щодо оформлення такого роду праць.

6. Недоліки кваліфікаційної роботи бакалавра

Кваліфікаційна робота виглядала б привабливіше, якби у ній були окреслені питання удосконалення методу кластеризації маршруту руху суден.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка ***«відмінно»****.*

Рецензент професор, д.т.н., професор кафедри комп'ютерної інженерії та інформаційних систем ХНУ

Лисенко Сергій Миколайович

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу бакалавра

студента *гр. КН-18-1 Боровика Дмитра Олеговича*
за темою: *Метод оцінки відповідності маршрутів кораблів до еталонного маршруту*

1. Актуальність теми

Актуальним завданням, яке потребує аналізу і досліджується у даній роботі, є визначення величини метрики, що застосовується для ідентифікації подібності маршрутів суден. Для ефективного використання автоматизованих систем необхідно передбачити застосування програмного модуля, який би дозволяв формувати висновки про наявність фактів порушення морського законодавства порівнюючи маршрути кораблів з еталонними маршрутами. Розробка такого метода є актуальною задачею комп'ютерних наук.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

За стандартом, а саме описом предметної області, об'єктами вивчення та діяльності є математичні, інформаційні, імітаційні моделі реальних явищ, об'єктів, систем і процесів та методи і технології отримання, зберігання, обробки, передачі та використання інформації. Метою роботи саме є розробка методу порівняння відповідності маршрутів суден відносно еталонного маршруту та розробка інформаційної системи реалізації вказаного методу. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробці інформаційних технологій. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

3. Професійні та особистісні якості бакалавра

При роботі над кваліфікаційною роботою бакалавра Боровик Дмитро Олегович проявив себе кваліфікованим фахівцем та дисциплінованим студентом, вчасно виконуючи поставлені етапи дослідження. Як в процесі написання пояснювальної записки, так і при розробці прикладного програмного забезпечення проявив достатні для одержання успішного результату компетентності та результати навчання. Опанував професійні скіли за напрямком «Комп'ютерні науки» та достатньо значний софт скіл .

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

5. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи показав достатній рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Тема роботи в повній мірі обґрунтована й розкрита, проведено аналіз актуальності та відомих досліджень в межах обраної теми, поставлені завдання, які у

роботі виконані, та розроблено програмне забезпечення для валідації та верифікації запропонованого метода.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Структура роботи та послідовність викладення логічні та відповідають поставленій меті. Викладення матеріалу послідовне, аргументоване, літературно грамотне.

8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Розроблений у роботі метод та його програмна реалізація може бути використана прикордонниками для підвищення ефективності виявлення порушників прикордонного законодавства.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

*Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «**відмінно»** .*

Керівник _____ *д.т.н., проф. зав. каф. КН Олександр БАРМАК*