

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Кафедра інформатики та програмної інженерії
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ Едуард ЖАРІКОВ
(підпис) (ім'я прізвище)

“ ____ ” _____ 2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інженерія програмного забезпечення
комп'ютерних систем»

спеціальності «121 Інженерія програмного забезпечення»

на тему: Веб-застосунок для моніторингу пошкодженого майна в Україні

Виконав студент IV курсу, групи ІТ-93
(шифр групи)

Ісмаїлов Расім Тельманович
(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник ст. викладач Вітковська І. І.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант
з графічної
документації

ст. викладач Головченко М. М.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Рецензент проф., д.т.н., проф. каф. ОТ Писарчук О. О.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

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

Студент _____
(підпис)

Київ –2023

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення
комп'ютерних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Едуард ЖАРІКОВ
(підпис) (ім'я прізвище)

“ ____ ” _____ 2023 р.

ЗАВДАННЯ
на дипломний проєкт студенту

Ісмаїлову Расіму Тельмановичу
(прізвище, ім'я, по батькові)

1. Тема проєкту Веб-застосунок для моніторингу пошкодженого майна в Україні

керівник проєкту Вітковська Ірина Іванівна, ст. викладач
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «31» травня 2023 р. №2101-с

2. Термін подання студентом проєкту « 17 » червня 2023 року

3. Вихідні дані до проєкту: технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: проведено постановку задачі, розглянуто наявні методи реалізації подібних задач, проведено огляд відомих технічних рішень, були розроблені функціональні і нефункціональні вимоги.

2) Моделювання та конструювання програмного забезпечення: було здійснено опис системи та її архітектури. Було також наведено список використаних технологій. Шляхом аналізу були прийняті рішення щодо проєктування і розробки застосунку.

3) Аналіз якості та тестування програмного забезпечення: було проведено тестування програмного забезпечення на відповідність вимогам.

4) Впровадження та супровід програмного забезпечення: наведено процес розгортання програмного забезпечення.

5. Перелік графічного матеріалу

1) Схема структурна варіантів використань _____

2) Схема бази даних _____

3) Креслення вигляду екранних форм _____

6. Консультанти розділів проєкту

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

7. Дата видачі завдання «10» березня 2023 року _____

Календарний план

| № з/п | Назва етапів виконання дипломного проєкту | Термін виконання етапів проєкту | Примітка |
|-------|---|---------------------------------|----------|
| 1 | Вивчення рекомендованої літератури | 10.03.2023 | |
| 2 | Аналіз існуючих методів розв'язання задачі | 21.04.2023 | |
| 3 | Постановка та формалізація задачі | 24.04.2023 | |
| 4 | Розробка інформаційного забезпечення | 20.05.2023 | |
| 5 | Алгоритмізація задачі | 10.04.2023 | |
| 6 | Обґрунтування вибору використаних технічних засобів | 18.05.2023 | |
| 7 | Розробка програмного забезпечення | 20.05.2023 | |
| 8 | Налагодження програми | 25.05.2023 | |
| 9 | Виконання графічних документів | 27.05.2023 | |
| 10 | Оформлення пояснювальної записки | 29.05.2023 | |
| 11 | Подання ДП на попередній захист | 02.06.2023 | |
| 12 | Подання ДП рецензенту | 12.06.2023 | |
| 13 | Подання ДП на основний захист | 17.06.2023 | |

Студент

(підпис)

Расім ІСМАІЛОВ

(ініціали, прізвище)

Керівник

(підпис)

Ірина ВІТКОВСЬКА

(ініціали, прізвище)

АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з чотирьох розділів, містить 38 таблиць, 18 рисунків та 7 джерел – загалом 64 сторінки.

Дипломний проєкт присвячений розробці та реалізації веб-застосунку для моніторингу пошкодженого майна в Україні.

Метою роботи є забезпечення зручного та ефективного способу інформування зацікавлених користувачів про випадки обстрілів і пошкоджень майна на території України.

Об'єкт дослідження: програмне забезпечення для надання та додавання інформації про випадки обстрілів.

Предмет дослідження: процес розробки програмного забезпечення для моніторингу пошкодженого майна.

У розділі «Аналіз вимог до програмного забезпечення» було проведено детальний аналіз предметної області, досліджено існуючі рішення. Було сформовано вимоги до майбутнього веб-застосунку. Також було представлено варіанти використання.

У розділі «Моделювання та конструювання програмного забезпечення» розглянуто використання застосунку авторизованими і неавторизованими користувачами. В розділі присутній опис архітектури програмного забезпечення.

У розділі «Аналіз якості та тестування програмного забезпечення» проведено тестування застосунку на відповідність очікуваній поведінці.

У розділі «Впровадження та супровід програмного забезпечення» було описано розгортання застосунку.

Програмне забезпечення впроваджено локально.

КЛЮЧОВІ СЛОВА: ВЕБ-ЗАСТОСУНОК, ASP.NET, ПОДІЯ, ОБСТРІЛ, БАЗА ДАНИХ, SVG, MVC, МОНІТОРИНГ.

ABSTRACT

The explanatory note of the diploma project consists of four sections, contains 37 tables, 17 figures and 7 sources - a total of 60 pages.

The diploma project is devoted to the development and implementation of a web application for monitoring damaged property in Ukraine.

The purpose of the work is to provide a convenient and effective way of informing interested users about cases of shelling and damage to property on the territory of Ukraine.

Research object: software for providing and adding information about shelling cases.

The subject of the study: the process of developing software for monitoring damaged property.

In the "Analysis of software requirements" section, a detailed analysis of the subject area was carried out, existing solutions were investigated. The requirements for the future web application were formed. Use cases were also presented.

The section "Software modeling and design" discusses the use of the application by authorized and unauthorized users. The section contains a description of the software architecture.

In the "Software Quality Analysis and Testing" section, the application is tested for compliance with the expected behavior.

The "Software Implementation and Maintenance" section described the deployment of the application.

The software is implemented locally.

KEYWORDS: WEB APPLICATION, ASP.NET, EVENT, FIRING, DATABASE, SVG, MVC, MONITORING.

| № з/п | Формат | Позначення | Найменування | Кількість листів | Примітка |
|-------|--------|--------------------------|---|---------------------|----------|
| 1 | A4 | | Завдання на дипломний проєкт | 2 | |
| 2 | A4 | КПІ.ІТ-9312.045440.01.91 | Технічне завдання | 18 | |
| 3 | A4 | КПІ.ІТ-9312.045440.02.81 | Пояснювальна записка | 60 | |
| 4 | A4 | КПІ.ІТ-9312.045440.03.12 | Текст програми | 47 | |
| 5 | A4 | КПІ.ІТ-9312.045440.04.51 | Програма та методика тестування | 6 | |
| 6 | A4 | КПІ.ІТ-9312.045440.05.34 | Керівництво користувача | 14 | |
| 7 | A3 | КПІ.ІТ-9312.045440.06.99 | Схема структурна варіантів використань | 1 | |
| 8 | A3 | КПІ.ІТ-9312.045440.08.99 | Схема бази даних | 1 | |
| 9 | A3 | КПІ.ІТ-9312.045440.10.99 | Креслення вигляду екранних форм | 1 | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

| | | | | | | | | |
|----------|------|------------------|-------|------|------------------------------|--|-------|---------|
| | | | | | КПІ.ІТ-9312.045440.00.90 | | | |
| Змін | Арк. | № докум. | Підп. | Дата | Відомість дипломного проєкту | Літ. | Аркуш | Аркушів |
| Розроб. | | Ісмаїлов Р. Т. | | | | | 1 | |
| Перевір. | | Вітковська І. І. | | | | | | |
| Н.контр. | | Ліщук К.І. | | | | | | |
| Затв. | | Жаріков Е.В. | | | | | | |
| | | | | | | КПІ ім. Ігоря Сікорського ФІОТ каф. ІІІ гр. ІТ-93 | | |

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ” _____ 2023 р.

Веб-застосунок для моніторингу пошкодженого майна в Україні

Технічне завдання

КПІ.ІТ-9312.045440.01.91

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Ірина Вітковська

Нормоконтроль:

_____ Максим Головченко

Виконавець:

_____ Расім Ісмаїлов

Київ – 2023

ЗМІСТ

| | | |
|-------|---|----|
| 1 | НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ | 4 |
| 2 | ПІДСТАВА ДЛЯ РОЗРОБКИ | 5 |
| 3 | ПРИЗНАЧЕННЯ РОЗРОБКИ | 6 |
| 4 | ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 7 |
| 4.1 | Вимоги до функціональних характеристик | 7 |
| 4.1.1 | Користувацького інтерфейсу | 7 |
| 4.1.2 | Для користувача: | 13 |
| 4.1.3 | Для адміністратора системи (якщо він передбачений): | 13 |
| 4.1.4 | Додаткові вимоги: | 13 |
| 4.2 | Вимоги до надійності | 13 |
| 4.3 | Умови експлуатації | 13 |
| 4.3.1 | Вид обслуговування | 14 |
| 4.3.2 | Обслуговуючий персонал | 14 |
| 4.4 | Вимоги до складу і параметрів технічних засобів | 14 |
| 4.5 | Вимоги до інформаційної та програмної сумісності | 14 |
| 4.5.1 | Вимоги до вхідних даних | 14 |
| 4.5.2 | Вимоги до вихідних даних | 15 |
| 4.5.3 | Вимоги до мови розробки | 15 |
| 4.5.4 | Вимоги до середовища розробки | 15 |
| 4.5.5 | Вимоги до представленню вихідних кодів | 15 |
| 4.6 | Вимоги до маркування та пакування | 15 |
| 4.7 | Вимоги до транспортування та зберігання | 15 |
| 4.8 | Спеціальні вимоги | 15 |
| 5 | ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ | 16 |

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 2 |

| | | |
|-----|--|----|
| 5.1 | Попередній склад програмної документації | 16 |
| 5.2 | Спеціальні вимоги до програмної документації | 16 |
| 6 | СТАДІЇ І ЕТАПИ РОЗРОБКИ | 17 |
| 7 | ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ | 18 |

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 3 |

1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: **Веб-застосунок для моніторингу пошкодженого майна в Україні**

Галузь застосування:

Наведене технічне завдання поширюється на розробку програмного забезпечення «Веб-застосунок для моніторингу пошкодженого майна в Україні» КП.ІТ-9312.045440.02.91, котре використовується для огляду і пошуку випадків пошкодження майна в Україні внаслідок військових дій, а також для подачі заявок на додавання нової інформації про ці події та їх подальшого збереження чи відхилення.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 4 |

2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки «Веб-застосунку для моніторингу пошкодженого майна в Україні» є завдання на дипломне проектування, затверджене кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 5 |

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для громадян, які бажають знати інформацію про випадки обстрілів в різних областях України. Також розробка може бути корисною для волонтерів, благодійних фондів та оперативних служб.

Метою розробки є надання зручного та простого інтерфейсу користувачу для ознайомлення з випадками обстрілів, їх пошуку за параметрами чи ключовими словами, а також для повідомлення про такі випадки власноруч.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 6 |

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

4.1.1 Користувацького інтерфейсу

– Авторизація в системі.

Неавторизований користувач натискає на кнопку особистого кабінету, яка представлена на рисунку 4.1 у вигляді кружка.



Рисунок 4.1 – Макет вигляду кнопки особистого кабінету

Він буде направлений на сторінку авторизація, яка показана на рисунку

4.2.

A mockup of a login form. It features the title 'Log in' at the top. Below the title are two input fields: 'Email' and 'Password'. Under the 'Password' field is a checkbox labeled 'Remember me?'. At the bottom of the form is a prominent blue button labeled 'Log in'. Below the button is a blue hyperlink that reads 'Register as a new user'.

Рисунок 4.2 – Макет сторінки введення даних авторизації

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 7 |

Після успішного вводу даних, користувач буде направлений на головну сторінку.

– Реєстрація в системі.

Незареєстрований користувач натискає на кнопку реєстрації на сторінці авторизації і потрапляє на сторінку реєстрації, яка представлена на рисунку 4.3.

Register

The image shows a registration form titled "Register". It consists of seven input fields stacked vertically, followed by a blue button labeled "Register". The fields are labeled: Name, Surname, Address, Age, Email, Password, and Confirm Password.

Рисунок 4.3 – Макет сторінка реєстрації

В разі успішної реєстрації, користувач потрапить на головну сторінку.

– Пошук подій за параметрами.

Користувач, навіть неавторизований, натискає на кнопку відкриття меню пошуку за параметрами, яка виглядає, як три вертикальні лінії з прозорими кружками в різних місцях. Кнопку показано на рисунку 4.4.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 8 |

Рисунок 4.4 – Макет кнопки відкриття меню пошуку за параметрами

При натисканні, користувач побачить меню вибору параметрів пошуку, яке показано на рисунку 4.5.

Select building type:

- Residential
- Industrial
- Administrative
- Healthcare
- School or childcare
- Military
- Commercial
- Religious
- Cultural
- Transport
- Humanitarian
- Food Infrastructure

Select weapon system:

- HE rocket artillery
- Cluster munition
- Incendiary munition
- Cruise missile
- Ballistic missile
- Thermobaric munition
- Vehicle mounted weapon
- Small arms
- Land mines
- Anti air missile
- Loitering munition

Date:




Рисунок 4.5 – Макет вигляду меню вибору параметрів пошуку

Користувач може вибрати типи будівель, інформацію по пошкодженню яких він хоче побачити; тип зброї, використаної для обстрілу; вибрати, за який день переглянути події; уточнити місце подій. Коли всі бажані параметри обрано, користувач натискає на кнопку пошуку. Меню вибору параметрів змінюється списком знайдених подій, який виглядає, як показано на рисунку 4.6.



Рисунок 4.6 – Макет вигляд списку подій, знайдених за обраними параметрами

– Пошук подій за ключовими словами.

Користувач вводить запит в поле для вводу, яке показано на рисунку 4.7.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 10 |



Рисунок 4.7 – Макет поле для пошуку по ключовим словам

Коли пошуковий запит введено, користувач натискає на кнопку пошуку, як на рисунку 4.7, справа.

– Відправка інформації про пошкоджене майно авторизованим користувачем.

Користувач, який успішно авторизувався в системі, натискає на кнопку додавання події, яка виглядає, як показано на рисунку 4.8.




Рисунок 4.8 – Макет вигляду кнопки додавання події

Після натискання, користувач потрапить на сторінку заповнення форми з інформацією про подію. Сторінка заповнення форми представлена на рисунку 4.9.

Add event

Event date:

Send

Рисунок 4.9 – Макет сторінки для додавання інформації про подію

В разі не заповнення необхідних полів, користувач не зможе відправити форму і побачить повідомлення про необхідність заповнити поля інформацією, як на рисунку 4.10.

Location

! Please fill out this field.

Description

Рисунок 4.10 – Макет повідомлення про необхідність заповнення обов'язкових полів

В разі коректного заповнення всіх необхідних полів, користувач може відправити форму на розгляд менеджерам застосунку.

4.1.2 Для користувача:

- реєстрація в системі;
- авторизація в системі;
- пошук подій за параметром «час»;
- пошук подій за параметром «тип використаної зброї»;
- пошук подій за параметром «тип враженої будівлі»;
- пошук подій за ключовими словами;
- редагування даних в особистому кабінеті;
- відправка повідомлення про зруйноване чи пошкоджене майно;

4.1.3 Для менеджера системи:

- авторизація в системі;
- пошук подій за параметром «час»;
- пошук подій за параметром «тип використаної зброї»;
- пошук подій за параметром «тип враженої будівлі»;
- пошук подій за ключовими словами;
- редагування даних в особистому кабінеті;
- обробка повідомлення про зруйноване чи пошкоджене майно.

4.1.4 Додаткові вимоги:

- розробити програмне забезпечення за типом веб-застосунку;
- серверна частина має бути реалізована на мові програмування C#;
- використати в якості бази даних SQL Server Express.

4.2 Вимоги до надійності

Передбачити контроль введення інформації. Передбачити захист від некоректних дій користувача. Забезпечити цілісність інформації в базі даних.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 13 |

4.3 Умови експлуатації

Умови експлуатації згідно СанПін 2.2.2.542 – 96.

4.3.1 Вид обслуговування

Вимоги до виду обслуговування не висуваються.

4.3.2 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не висуваються.

4.4 Вимоги до складу і параметрів технічних засобів

Програмне забезпечення повинно працювати на персональних комп'ютерах або серверах з операційною системою Windows/Linux, які підтримують сучасні версії браузерів.

Мінімальна конфігурація технічних засобів:

- тип процесору: Intel Core i5;
- об'єм ОЗП: 4 Гб;
- підключення до мережі Інтернет зі швидкістю від 20 мегабіт.

Рекомендована конфігурація технічних засобів :

- тип процесору: Intel Core i5;
- об'єм ОЗП: 8 Гб;
- підключення до мережі Інтернет зі швидкістю від 100 мегабіт.

4.5 Вимоги до інформаційної та програмної сумісності

Програмне забезпечення повинно працювати під управлінням операційних систем Windows/Linux.

4.5.1 Вимоги до вхідних даних

Вхідні дані повинні бути представлені в наступному форматі: HTTP-запит до серверу.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 14 |

4.5.2 Вимоги до вихідних даних

Результати повинні бути представлені в наступному форматі: HTTP-відповідь серверу.

4.5.3 Вимоги до мови розробки

Розробку виконати на мові програмування C#.

4.5.4 Вимоги до середовища розробки

Розробку виконати на платформі Windows 10, за допомогою JetBrains Rider.

4.5.5 Вимоги до представлення вихідних кодів

Вихідний код програми має бути представлений у вигляді файлів розширення .cs, .cshtml у документі Текст Програми.

4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

4.8 Спеціальні вимоги

Згенерувати інсталяційну версію програмного забезпечення.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 15 |

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Попередній склад програмної документації

У склад супроводжувальної документації повинні входити наступні документи на аркушах формату А4:

- пояснювальна записка;
- технічне завдання;
- текст програми;
- програма та методика тестування;
- керівництво користувача;
- керівництво менеджера;

Графічна частина повинна бути виконана на аркушах формату А3 та містити наступні документи:

- схема структурна варіантів використання;
- схема бази даних;
- креслення вигляду екранних форм;

5.2 Спеціальні вимоги до програмної документації

Спеціальні вимоги до програмної документації відсутні.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 16 |

6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

| № | Назва етапу | Строк | Звітність |
|----|---|-------|---|
| 1. | Вивчення літератури за тематикою проекту | 10.03 | |
| 2. | Розробка технічного завдання | 04.04 | Технічне завдання |
| 3. | Аналіз вимог та уточнення специфікацій | 05.04 | Специфікації програмного забезпечення |
| 4. | Проектування структури програмного забезпечення, проектування компонентів | 20.04 | Схема структурна програмного забезпечення та специфікація компонентів |
| 5. | Програмна реалізація програмного забезпечення | 30.04 | Тексти програмного забезпечення |
| 6. | Тестування програмного забезпечення | 10.05 | Тести, результати тестування |
| 7. | Розробка матеріалів текстової частини проекту | 15.05 | Пояснювальна записка |
| 8. | Розробка матеріалів графічної частини проекту | 20.04 | Графічний матеріал проекту |
| 9. | Оформлення технічної документації проекту | 24.05 | Технічна документація |

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 17 |

7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.01.91 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 18 |

**Пояснювальна записка
до дипломного проєкту**

на тему: Веб-застосунок для моніторингу пошкодженого майна в Україні

КПІ.ІТ-9312.045440.02.81

Київ – 2023

ЗМІСТ

| | |
|---|----|
| Вступ | 6 |
| 1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 8 |
| 1.1 Загальні положення | 8 |
| 1.2 Змістовний опис і аналіз предметної області | 8 |
| 1.3 Аналіз існуючих технологій та успішних ІТ-проектів | 9 |
| 1.4 Аналіз вимог до програмного забезпечення | 12 |
| 1.4.1 Розроблення функціональних вимог | 19 |
| 1.4.2 Розроблення нефункціональних вимог | 24 |
| 1.5 Постановка задачі | 24 |
| Висновки до розділу | 25 |
| 2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 26 |
| 2.1 Моделювання та аналіз програмного забезпечення | 26 |
| 2.2 Архітектура програмного забезпечення | 32 |
| 2.3 Конструювання програмного забезпечення | 34 |
| 2.4 Аналіз безпеки даних | 43 |
| Висновки до розділу | 44 |
| 3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 46 |
| 3.1 Аналіз якості ПЗ | 46 |
| 3.2 Опис процесів тестування | 46 |
| 3.3 Опис контрольного прикладу | 54 |
| Висновки до розділу | 59 |

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 2 |

| | | |
|-----|---|----|
| 4 | ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 60 |
| 4.1 | Розгортання програмного забезпечення | 60 |
| 4.2 | Робота з програмним забезпеченням | 60 |
| | Висновки до розділу | 61 |
| | ВИСНОВКИ | 62 |
| | СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 64 |
| | ДОДАТКИ | 65 |

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 3 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- BPMN** – Business Process Model and Notation – нотація для моделювання бізнес-процесів, що дозволяє візуалізувати послідовність та взаємодії між діями та об'єктами в бізнес-середовищі.
- DI** – Dependency Injection – підхід до розробки програмного забезпечення, де залежності між об'єктами передаються через зовнішній об'єкт або контейнер.
- ER** – Entity-Relationship – модель даних, що описує структуру бази даних за допомогою сутностей, атрибутів та відношень між ними.
- SVG** – Scalable Vector Graphics – масштабована векторна графіка.
- HTML** – HyperText Markup Language – мова розмітки для створення веб-сторінок та веб-додатків.
- HTTP** – Hypertext Transfer Protocol – протокол передачі гіпертекстових документів у мережі Інтернет.
- IT** – Інформаційні технології
- MVC** – Model-View-Controller – патерн проектування, що використовується для розділення логіки програми на модель (дані), представлення (інтерфейс) та контролер (логіка).
- IoC** – Inversion of Control – це принцип побудови програми, при якому її частини викликаються із загальної спільновикористовуваної бібліотеки.
- SQL** – Structured Query Language, мова структурованих запитів, використовується для керування та маніпуляції даними у базах даних.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 4 |

- UML – Unified Modeling Language – стандартна мова моделювання, що використовується для візуалізації, специфікації та документування архітектури програмних систем.
- БД – База даних – структуроване сховище даних, яке дозволяє зберігати, організовувати та керувати даними для подальшого використання.
- СКБД – Операційна система. Система керування базами даних – програмне забезпечення для управління, організації та збереження даних у базах даних.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 5 |

Вступ

Сьогодні Україна переживає один з найскладніших періодів своєї історії через війну. В результаті воєнних дій багато майна було пошкоджено та зруйновано, що призвело до значних втрат для населення та економіки.

«Збитки війни, як правило, виражаються через оцінку втрат, стертого економічного потенціалу та пошкодження найважливішої інфраструктури країни. За цими стандартами Київська школа економіки нині оцінює такі збитки в Україні у 80,4 мільярда доларів, при цьому щотижня вони збільшуються на суму від 8 до 12 мільярдів доларів. Якщо нинішня російська військова зосередженість на сході та півдні України призведе до вкоріненої та тривалої війни, ця цифра незабаром перевищить 135 мільярдів доларів нинішньої вартості плану Маршалла під керівництвом США, який допоміг відновити Європу після Другої світової війни, і 145 мільярдів доларів США на відбудову, витрачені в Афганістані.» [1]

У зв'язку з цим, дуже важливо мати інструмент для моніторингу та аналізу пошкодженого майна в Україні. Такий інструмент допоможе швидко та ефективно відновлювати зруйновану інфраструктуру, забезпечити допомогу постраждалим, тримати населення в курсі щодо ситуації в різних областях України.

У цій роботі пропонується розробка інструменту для моніторингу пошкодженого майна в Україні. Рішення являє собою веб-застосунок з картою, на якій візуально представлена обстановка в різних областях. У користувачів буде можливість заповнення певної веб форми, якщо постраждало їх майно, з послідуєчим додаванням цієї інформації на карту. Сайт матиме зручний інтерфейс.

На сьогоднішній день відновлення пошкодженого майна є однією з найважливіших проблем, з якими стикається країна. На жаль, інформація про пошкоджене майно часто недоступна або розрізняється між різними

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 6 |

джерелами. Тому потрібен інструмент, який забезпечить зручний та швидкий доступ до цієї інформації.

Розробка веб застосунку для моніторингу пошкодженого майна в Україні є важливим кроком в напрямку вирішення цієї проблеми.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 7 |

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Одним із негативних наслідків війни є пошкодження майна, а також вплив цих пошкоджень на громадян та інфраструктуру. Розробка застосунку для моніторингу пошкодженого майна базується на важливості отримання оперативної та зручної інформації, яка допоможе організаціям та окремим особам здійснювати ефективне планування та прийняття рішень.

Основна мета створення даного веб-застосунку – покращити доступ до інформації про руйнування внаслідок військових дій.

Задля реалізації поставленої мети, було створено веб-застосунок, який візуалізує інформацію на карті по окремих областях України. У користувачів також є змога створити і відправити форму з інформацією про випадки пошкодження майна, внаслідок війни з допомогою сайту.

1.2 Змістовний опис і аналіз предметної області

Війна завдає колосальних збитків, зокрема цивільній інфраструктурі, житлу та іншому рухомому і нерухомому майну.

Щоб визначити рівень руйнувань в окремих регіонах України, можна використати різні джерела інформації, такі як звіти державних органів, засоби масової інформації та інші.

Проте інформації про подібні випадки дуже багато, а нові події трапляються майже не щодня в різних областях країни. І виходить так, що ці дані часто розпорошені, неоднорідні та неструктуровані, що ускладнює їх аналіз та використання для моніторингу і прийняття рішень.

Ця ситуація створює потребу у розробці рішення для моніторингу пошкодженого майна, яке б значно полегшило процес збору, обробки та візуалізації даних про пошкодження.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 8 |

1.3 Аналіз існуючих технологій та успішних ІТ-проектів

Окрім різноманітних джерел, таких як звіти держаних органів та засобів масової інформації, існують і рішення у вигляді інтерактивних карт.

Наприклад, інтерактивна карта для відслідковування військових злочинів від інтернет видання «Bellingcat» (рис. 1.1) та «Карта руйнувань» (рис 1.2) від волонтерського руху «ReUkraine».

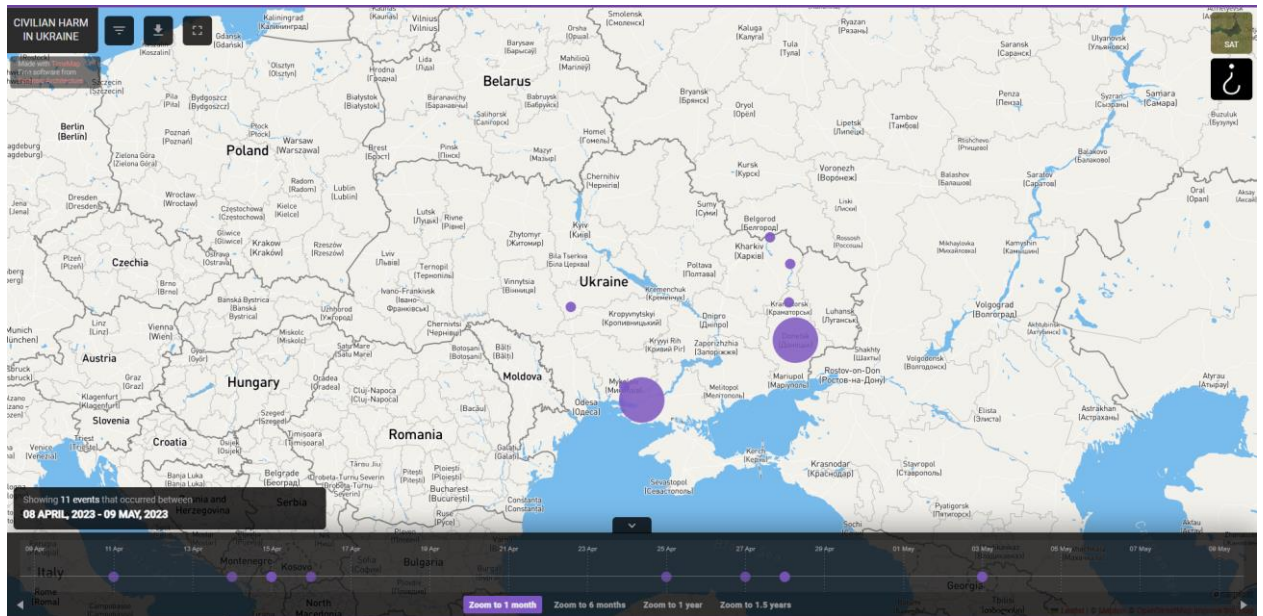


Рисунок 1.1 – Скріншот сайту ukraine.bellingcat.com

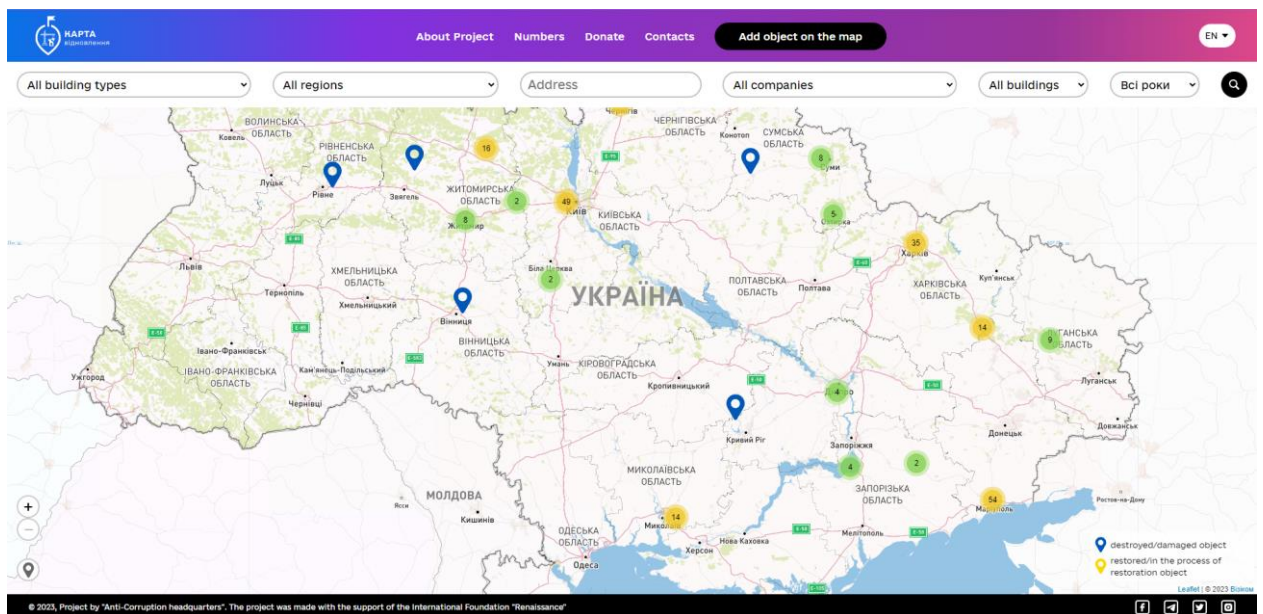


Рисунок 1.2 – Скріншот сайту reukraine.shtab.net

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
|-------|------|----------|-------|-------|

Обидва рішення доволі добре справляються з задачею фіксації цих подій та інформування населення. Кожен випадок описується місцем, датою, коротким описом наслідків, а також до багатьох з них прикріплене посилання на джерело інформації.

«Карта руйнувань» навіть показує фотографії до руйнувань і після, хоч і не для кожного такого випадку.

Проект від «Bellingcat» дозволяє переглядати події за конкретний рік, місяць чи день.

До недоліків існуючих рішень можна віднести:

- інколи неповну чи неактуальну інформацію. Карти можуть не відображати всі руйнування, що мають місце в реальності, або не оновлюватись достатньо часто, щоб відображати поточну інформацію;
- помилкова прив'язка до місця. Дані карти проекту ReUkraine інколи не відповідають реальності через помилки в зборі чи неправильну їх інтерпретацію. Як, наприклад, певна подія, що мала місце на території України, відображається за межами країни;
- відсутність джерела. В карті від «Bellingcat» є події, які посилаються на пост в Telegram. Але, якщо цей пост видалено, на місці джерела інформації можна спостерігати повідомлення про те, що пост було видалено.

Порівняння аналогів наведено в таблиці 1.1.

Таблиця 1.1 – Порівняння з аналогами

| Функціонал | Даний застосунок | Bellingcat | ReUkraine | Пояснення |
|--------------------------------|------------------|------------|-----------|---|
| Розширена інформація про подію | Так | Так | Так | Можливість дізнатись дату події, місце, джерело інформації. |

Продовження таблиці 1.1

| | | | | |
|--|-----|-----|-----|---|
| Узагальнена статистика по кількості подій в певній області | Так | Так | Так | Дає можливість оцінити, які області страждають від обстрілів частіше. |
| Інформація про тип зброї | Так | Так | Ні | Можливість дізнатись тип зброю, з якої було здійснено обстріл. |
| Можливість фільтрації по даті | Так | Так | Ні | Можливість перегляду подій, які стались в певний день. |
| Можливість фільтрації по типу пошкодженої споруди | Так | Так | Так | Можливість переглянути інформацію по видам інфраструктури, що піддається обстрілам частіше. |

Продовження таблиці 1.1

| | | | | |
|--------------------------|-----|----|----|---|
| Пошук по ключовим словам | Так | Ні | Ні | Пошук події по словам, які можуть зустрічатись в описі. |
|--------------------------|-----|----|----|---|

1.4 Аналіз вимог до програмного забезпечення

Визначимо перелік акторів та їх сценаріїв взаємодії (рис 1.3). Користувачі можуть мати одну з наведеного нижче переліку ролей:

- неавторизований користувач;
- авторизований користувач;
- менеджер.

Після успішної реєстрації або авторизації, неавторизований користувач отримує роль авторизованого користувача системи, що розширює список можливостей використання застосунку.

Авторизований користувач може:

- вносити дані про себе у профіль;
- створювати та відправляти форми з інформацією про пошкодження майна.

Роль менеджера передбачає:

- перегляд заповнених форм від авторизованих користувачів;
- можливість відхилення або підтвердження форм, отриманих від користувачів;
- можливість взаємодії з авторизованими користувачами.

При проведенні аналізу вимог до застосунку, було виділено ряд вимог до різних аспектів застосунку.

Картографічний функціонал:

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 12 |

- відображення карти з позначеними різними кольорами областями в залежності від масштабів руйнувань;

- можливість фільтрування даних про руйнування за часом, місцем.

Функціонал для реєстрації та внесення даних:

- можливість реєстрації та сайті та авторизації користувачів;

- можливість внесення даних про руйнування в різних областях

України;

- обмеження прав доступу до внесення даних в залежності від рівня користувача (менеджер, авторизований користувач, неавторизований користувач).

Захист даних:

- захист персональних даних користувачів;

- захист даних, що внесені на сайт користувачами від несанкціонованого доступу та зміни.

Інтерфейс:

- зручний та інтуїтивно зрозумілий інтерфейс для користувачів.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 13 |

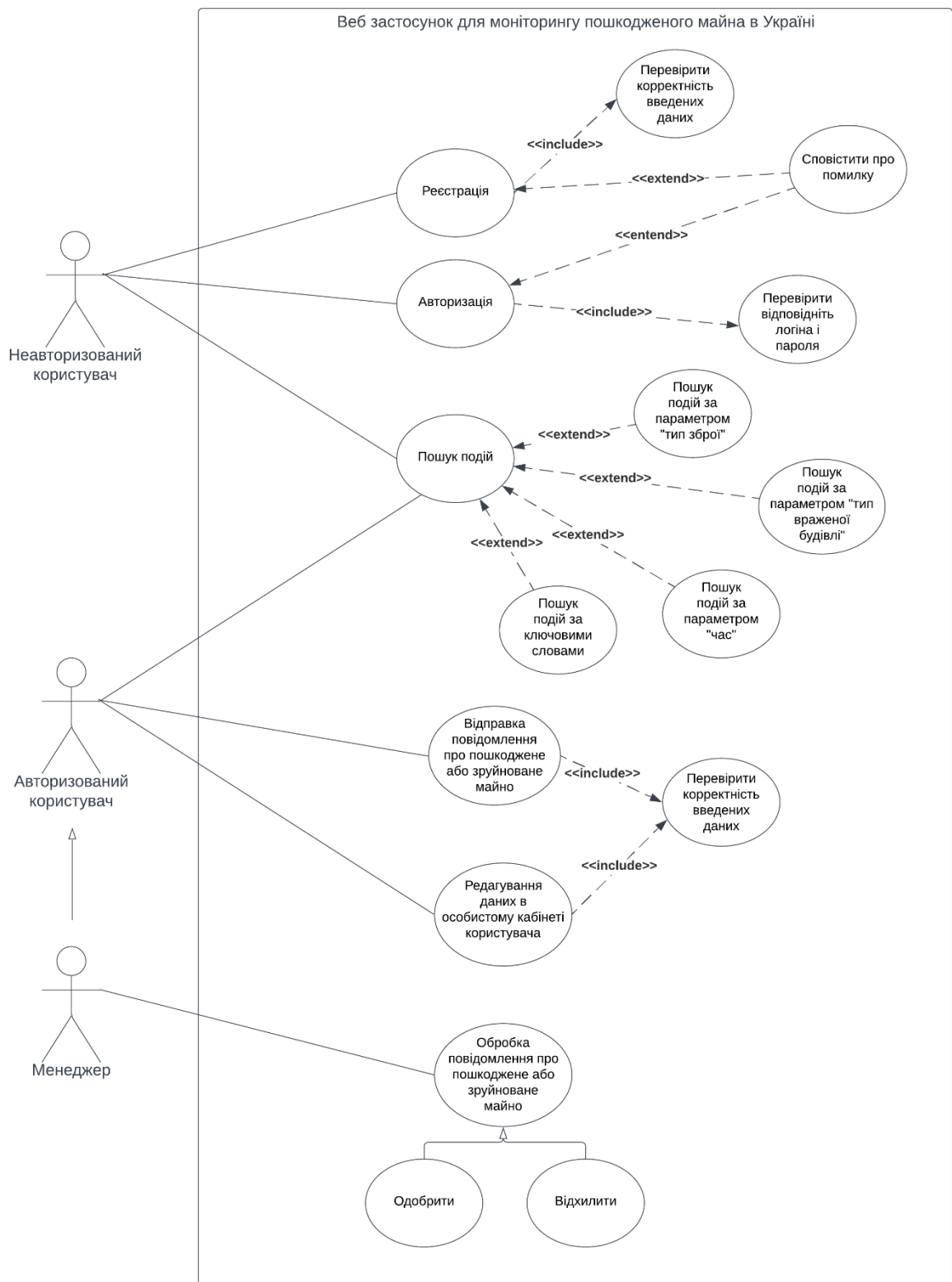


Рисунок 1.3 – Діаграма варіантів використання

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
| | | | | |

В таблицях 1.2 - 1.17 наведені варіанти використання програмного забезпечення.

Таблиця 1.2 - Варіант використання UC-01

| | |
|----------------|--|
| Use case name | Реєстрація користувача. |
| Use case ID | UC-01 |
| Goals | Реєстрація нового користувача в системі. |
| Actors | Незареєстрований користувач. |
| Trigger | Користувач бажає зареєструватися. |
| Pre-conditions | Користувач знаходиться на сторінці авторизації. |
| Flow of Events | Користувач натискає кнопку реєстрації. На сторінці реєстрації, в поля для реєстрації вводяться необхідні для реєстрації інформація. Після заповнення даних користувач натискає кнопку підтвердження реєстрації. Після цього з'являється повідомлення про успішну реєстрацію, і користувач перенаправляється на сторінку входу. |
| Extension | В випадку введення не коректних даних, користувач побачить повідомлення про невідповідність введених даних очікуваним і не зможе відправити форму реєстрації. |
| Post-Condition | Створення сторінки користувача, перехід на сторінку входу. |

Таблиця 1.3 - Варіант використання UC-02

| | |
|---------------|--------------------------|
| Use case name | Авторизація користувача. |
| Use case ID | UC-02 |

Продовження таблиці 1.3

| | |
|----------------|--|
| Goals | Авторизація користувача в системі. |
| Actors | Неавторизований користувач. |
| Trigger | Користувач бажає авторизуватись. |
| Pre-conditions | - |
| Flow of Events | Користувач переходить на сторінку авторизації. Вносить дані для авторизації. Після заповнення необхідних полів, натискає кнопку авторизації. Користувач потрапляє на головну сторінку. |
| Extension | В випадку введення не коректних даних, користувач побачить повідомлення про невідповідність введених даних очікуваним і не зможе відправити форму авторизації. Якщо користувач з таким логіном не зареєстрований, чи зареєстрований, але був введений неправильний пароль, користувач побачить повідомлення з помилкою авторизації. |
| Post-Condition | Авторизація в системі, перехід на головну сторінку. |

Таблиця 1.4 - Варіант використання UC-03

| | |
|---------------|---|
| Use case name | Пошук подій. |
| Use case ID | UC-03 |
| Goals | Пошук подій по заданому критерію. |
| Actors | Неавторизований користувач, авторизований користувач. |
| Trigger | Користувач бажає переглянути події, які відповідають заданому критерію. |

Продовження таблиці 1.4

| | |
|----------------|--|
| Pre-conditions | - |
| Flow of Events | Користувач відкриває меню вибору критеріїв пошуку, обирає бажаний критерій, або декілька критеріїв та натискає кнопку підтвердження. |
| Extension | |
| Post-Condition | Користувач бачить інформацію про події, що відповідають обраним критеріям. |

Таблиця 1.5 - Варіант використання UC-04

| | |
|----------------|---|
| Use case name | Відправка повідомлення про зруйноване майно. |
| Use case ID | UC-04 |
| Goals | Відправити повідомлення про пошкоджене або зруйноване майно. |
| Actors | Авторизований користувач. |
| Trigger | Користувач бажає повідомити про подію. |
| Pre-conditions | - |
| Flow of Events | Користувач натискає кнопку повідомлення про пошкоджене чи зруйноване майно. Користувач потрапляє на сторінку з формою для заповнення. Користувач заповнює необхідні поля і натискає кнопку для відправки форми. |
| Extension | Якщо введені дані не відповідають очікуваному формату, користувач побачить повідомлення про помилку при заповненні і не зможе відправити форму. |

Продовження таблиці 1.5

| | |
|----------------|---|
| Post-Condition | Користувач потрапляє на головну сторінку. |
|----------------|---|

Таблиця 1.6 - Варіант використання UC-05

| | |
|----------------|--|
| Use case name | Редагування даних профілю. |
| Use case ID | UC-05 |
| Goals | Редагувати дані в особистому кабінеті. |
| Actors | Авторизований користувач. |
| Trigger | Користувач бажає змінити дані профілю. |
| Pre-conditions | - |
| Flow of Events | Користувач переходить в особистий кабінет та натискає кнопку редагування. Користувач вносить бажані зміни та натискає кнопку збереження. |
| Extension | Якщо введені дані не відповідають очікуваному формату, користувач побачить повідомлення про помилку і не зможе зберегти зміни. |
| Post-Condition | Користувач потрапляє на сторінку особистого кабінету. |

Таблиця 1.7 - Варіант використання UC-06

| | |
|---------------|---|
| Use case name | Обробка повідомлення про пошкоджене або зруйноване майно. |
| Use case ID | UC-06 |

Продовження таблиці 1.7

| | |
|----------------|---|
| Goals | Підтвердити чи відхилити повідомлення про пошкоджене або зруйноване майно. |
| Actors | Менеджер. |
| Trigger | Менеджер бажає переглянути повідомлення про події і допомогти в прийнятті рішення щодо них. |
| Pre-conditions | - |
| Flow of Events | Менеджер переходить на сторінку огляду подій, відправлених авторизованими користувачами щодо зруйнованого чи пошкодженого майна. Менеджер ознайомлюється з наданою інформацією і натискає одну з кнопок: для підтвердження чи для відхилення події. |
| Extension | - |
| Post-Condition | - |

1.4.1 Розроблення функціональних вимог

Функціональні вимоги – це вимоги до програмного забезпечення, що описують роботу системи, її поведінку: внесення даних, обробка даних та інші специфічні операції, які має робити система.

В процесу аналізу, в застосунку було виділено функціональні вимоги, описані таблицями 1.8-1.16.

Таблиця 1.8 – Опис функціональної вимоги RQ-01

| | |
|--------------------------|-----------------------|
| Унікальний ідентифікатор | RQ-01 |
| Назва | Реєстрація в системі. |

Продовження таблиці 1.8

| | |
|------|--|
| Опис | Система надає можливість користувачу надати необхідну інформацію, щоб зареєструватись і в подальшому набути ролі авторизованого користувача. |
|------|--|

Таблиця 1.9 – Опис функціональної вимоги RQ-02

| | |
|--------------------------|--|
| Унікальний ідентифікатор | RQ-02 |
| Назва | Авторизація в системі |
| Опис | Система надає можливість користувачу, який вже зареєструвався, авторизуватись. |

Таблиця 1.10 – Опис функціональної вимоги RQ-03

| | |
|--------------------------|---|
| Унікальний ідентифікатор | RQ-03 |
| Назва | Пошук подій за параметром «час». |
| Опис | Користувач знаходиться в меню вибору параметрів пошуку і обирає параметр, який відповідає за час події. |

Таблиця 1.11 – Опис функціональної вимоги RQ-04

| | |
|--------------------------|--|
| Унікальний ідентифікатор | RQ-04 |
| Назва | Пошук подій за параметром «тип зброї». |

Продовження таблиці 1.11

| | |
|------|---|
| Опис | Користувач знаходиться в меню вибору параметрів пошуку і обирає параметр, який відповідає за тип використаної під час обстрілу зброї. |
|------|---|

Таблиця 1.12 – Опис функціональної вимоги RQ-05

| | |
|--------------------------|---|
| Унікальний ідентифікатор | RQ-05 |
| Назва | Пошук подій за параметром «тип враженої будівлі». |
| Опис | Користувач знаходиться в меню вибору параметрів пошуку і обирає параметр, який відповідає за тип враженої під час обстрілу будівлі. |

Таблиця 1.13 – Опис функціональної вимоги RQ-06

| | |
|--------------------------|--|
| Унікальний ідентифікатор | RQ-06 |
| Назва | Пошук подій за ключовими словами. |
| Опис | Користувач натискає на кнопку пошуку подій за ключовими словами і вписує бажаний запит |

Таблиця 1.14 – Опис функціональної вимоги RQ-07

| | |
|--------------------------|--|
| Унікальний ідентифікатор | RQ-07 |
| Назва | Редагування даних в особистому кабінеті користувача. |
| Опис | Авторизований користувач заходить в особистий кабінет і вносить бажані зміни в своєму профілі. |

Таблиця 1.15 – Опис функціональної вимоги RQ-08

| | |
|--------------------------|---|
| Унікальний ідентифікатор | RQ-08 |
| Назва | Відправка повідомлення про зруйноване або пошкоджене майно. |
| Опис | Авторизований користувач натискає на кнопку, яка відповідає за відкриття сторінки для повідомлення про зруйноване чи пошкоджене майно і заповнює запропоновану форму. |

Таблиця 1.16 – Опис функціональної вимоги RQ-09

| | |
|--------------------------|---|
| Унікальний ідентифікатор | RQ-09 |
| Назва | Обробка отриманої форми з інформацією про руйнування чи пошкодження майна. |
| Опис | Модератор, ознайомившись з даними, обирає, віддати голос за підтвердження форми, чи за її відхилення. |

Відповідність варіантів використання функціональним вимогам представлена в таблиці 1.17.

Таблиця 1.17 – Матриця трасування вимог

| | RQ-01 | RQ-02 | RQ-03 | RQ-04 | RQ-05 | RQ-06 | RQ-07 | RQ-08 | RQ-09 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| UC-01 | | | | | | | | | |
| UC-02 | | | | | | | | | |

Продовження таблиці 1.17

| | | | | | | | | | |
|-------|--|--|--|--|--|--|--|--|--|
| UC-03 | | | | | | | | | |
| UC-04 | | | | | | | | | |
| UC-05 | | | | | | | | | |
| UC-06 | | | | | | | | | |

В таблиці 1.18 представлена модель вимог. Реєстрація, авторизація, відправка форми про зруйноване або пошкоджене майно та її обробка мають високий пріоритет. Вимоги, пов'язані з пошуком подій, відповідають середньому пріоритету, в той час як редагування даних в особистому кабінеті має низький пріоритет.

Таблиця 1.18 – Модель вимог

| Вимога | Код | Пріоритет | Ризик |
|--|-------|-----------|----------|
| Реєстрація в системі. | RQ-01 | Високий | Середній |
| Авторизація в системі | RQ-02 | Високий | Низький |
| Пошук подій за параметром «час». | RQ-03 | Середній | Низький |
| Пошук подій за параметром «тип зброї». | RQ-04 | Середній | Низький |
| Пошук подій за параметром «тип враженої будівлі». | RQ-05 | Середній | Низький |
| Пошук подій за ключовими словами. | RQ-06 | Середній | Низький |
| Редагування даних в особистому кабінеті користувача. | RQ-07 | Низький | Середній |
| Відправка повідомлення про зруйноване або пошкоджене майно. | RQ-08 | Високий | Середній |
| Обробка отриманої форми з інформацією про руйнування чи пошкодження майна. | RQ-09 | Високий | Середній |

1.4.2 Розроблення нефункціональних вимог

До нефункціональних вимог можна віднести наступні:

- кросплатформність: забезпечити доступ до веб -застосунку з широкого кола пристроїв, які мають підключення до інтернету;
- зручний інтерфейс: забезпечити користувачам зручний інтерфейс, щоб вони могли швидко і легко знайти потрібну інформацію або додати свою;
- певний рівень захисту даних користувачів;
- сумісність з .NET Core 3.1.

1.5 Постановка задачі

Відповідно до функціональних і нефункціональних вимог, виділимо ряд завдань, що має вирішувати розроблюване програмне рішення:

- можливість оцінки ситуації в окремих областях з допомогою картографічного представлення;
- можливість пошуку подій за ключовими словами;
- можливість пошуку подій по певним параметрам;
- можливість реєстрації;
- можливість авторизації;
- можливість повідомляти про події менеджерів;
- можливість підтверджувати інформацію від користувачів менеджерами;
- можливість відхиляти інформацію від користувачів менеджерами.

Ряд завдань, які вирішуються модулем, є ключовими для функціонування системи.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 24 |

Висновки до розділу

В даному розділі було проведено постановку задачі і оглянуто основні проблеми, пов'язані з моніторингом пошкодженого майна в Україні. Також були розглянуті наявні методи реалізації подібних задач і виявлені їх обмеження.

Також проведено огляд відомих технічних рішень та підходів, проведено їх порівняння, виділення переваг та недоліків і встановлено, що існують способи змінити чи покращити систему.

В результаті була розроблена система функціональних та нефункціональних вимог, що встановлюють потрібний функціонал і якість системи. Крім того, було поставлено комплекс завдань для розроблюваного модулю згідно з цими вимогами. В цілому, цей розділ надав загальний огляд проблеми, провів порівняння з існуючими рішеннями і сформулював вимоги до розроблюваної системи.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 25 |

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Для проведення аналізу та моделювання програмного забезпечення, розглянемо BPMN діаграми, які надають опис основних сценаріїв виконання веб-застосування.

Розглянемо процес реєстрації користувача в системі:

- користувач натискає на кнопку авторизації на головній сторінці;
- користувач натискає кнопку реєстрації;
- користувач заповнює необхідні поля форми такими даними: адреса електронної пошти, ім'я, прізвище, по батькові, пароль, підтвердження паролю;
- перевіряється коректність введеної інформації;
- у разі невідповідності введених даних тим, які очікуються в певних полях - користувач бачить сповіщення про помилку та виправляє дані;
- користувач натискає кнопку підтвердження відправки форми;
- після отримання даних сервером, перевіряється унікальність адреси електронної пошти;
- якщо введена адреса пошти вже прив'язана до існуючого профілю, користувачу буде запропоновано авторизуватись;
- якщо всі етапи пройдено і записів про користувачів з таким адресом електронної пошти на сервері немає – буде створено новий профіль користувача з даними, вказаними в отриманій формі реєстрації. Запис буде збережено в базі даних, а користувача буде направлено на головну сторінку.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 26 |

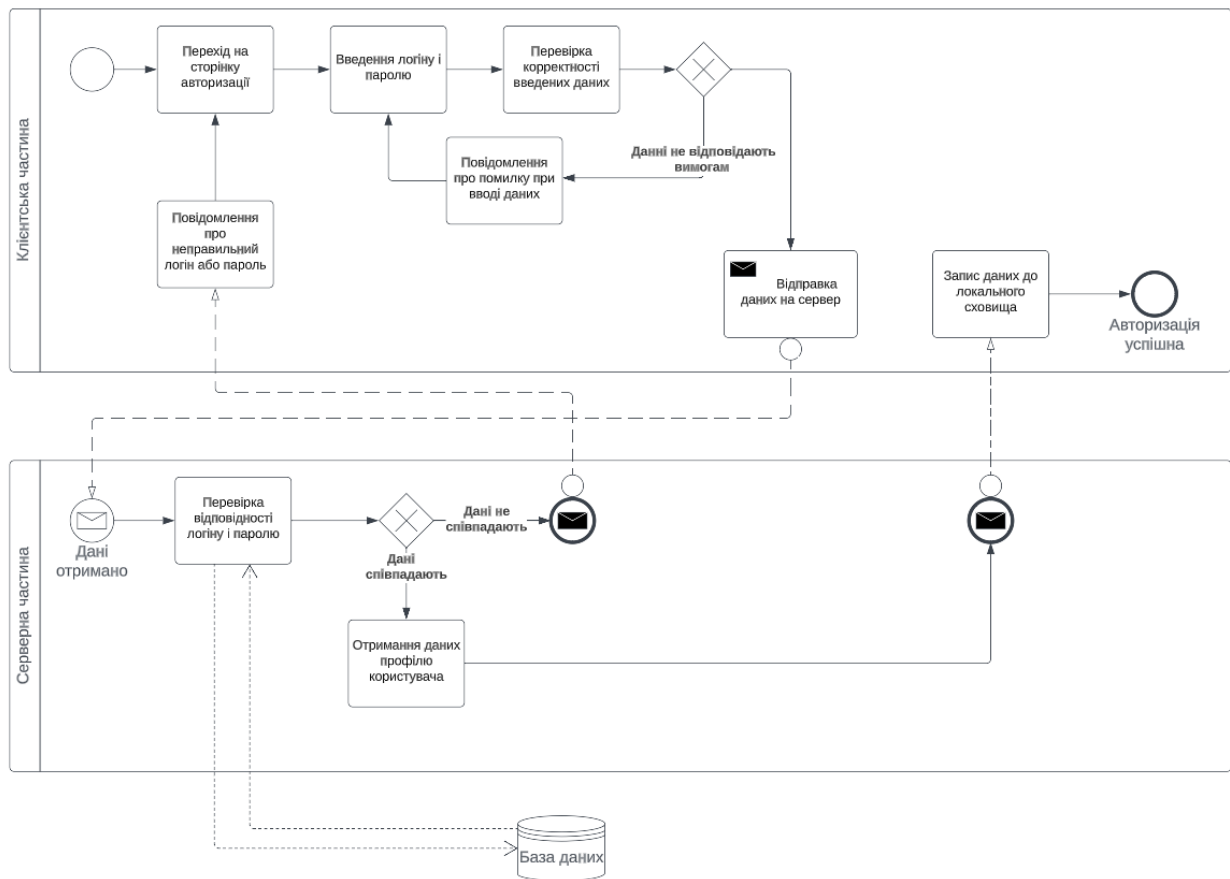


Рисунок 2.2 – Схема бізнес-процесу авторизації користувача

Опис запити на пошук інформації по заданому параметру або декільком параметрам параметру:

- користувач, знаходячись на головній сторінці застосунку, натискаю на кнопку вибору параметрів для пошуку;
- користувач відмічає необхідні параметри (один або декілька) і натискає на кнопку пошуку;
- серверна частина завантажує інформацію з бази даних, відповідно до заданих критеріїв;
- дані розбиваються відповідно до області і відправляються для подальшого відображення;
- отримана інформація про знайдені події відображається користувачу.

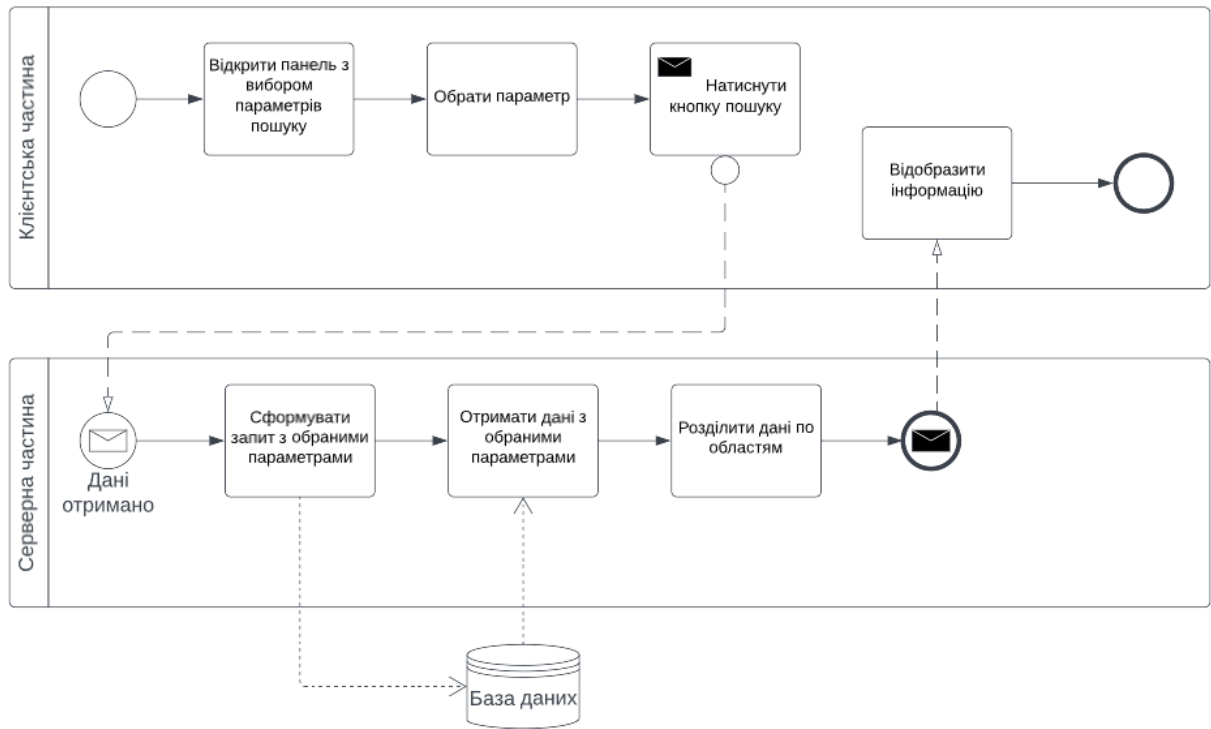


Рисунок 2.3 Схема бізнес-процесу пошуку подій за критерієм

Розглянемо процес зміни інформації в профілі користувача:

- авторизований користувач заходить в особистий кабінет;
- користувач натискає на кнопку редагування профілю;
- вносить бажані зміни;
- якщо введені дані не в коректному форматі, користувач побачить повідомлення з помилкою;
- натискає кнопку збереження змін;
- передати повідомлення про виконану операцію клієнту.

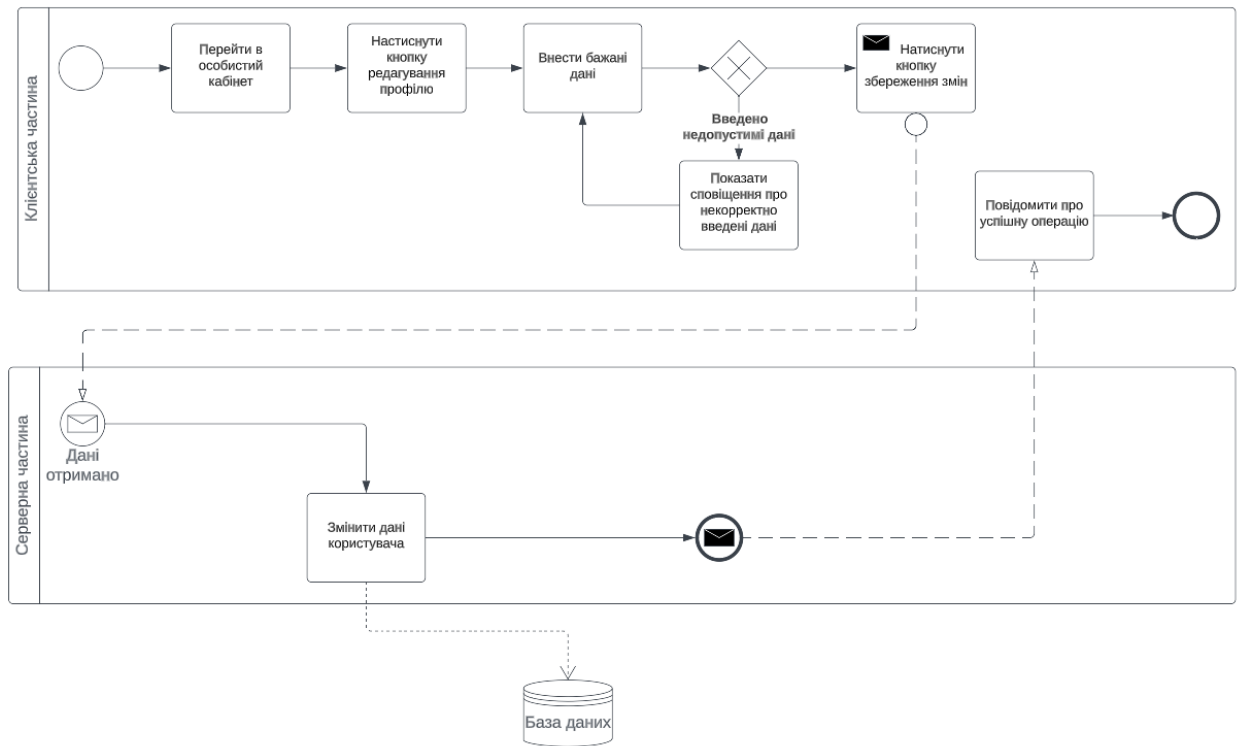


Рисунок 2.4 – Схема бізнес процесу зміни персональних даних авторизованим користувачем

Процес повідомлення про зруйноване або пошкоджене майно:

- користувач натискає на кнопку сповіщення про зруйноване чи пошкоджене майно;
- якщо користувач не авторизований – він потрапить на сторінку авторизації;
- авторизований же користувача буде направлено на сторінку з формою для заповнення;
- користувач вводить необхідні дані ;
- якщо дані не відповідають очікуваному формату – користувач побачить повідомлення про некоректність введених даних;
- у випадку заповнення форми без помилок, інформація буде відправлена на сервер;
- дані зберігаються в базу даних.

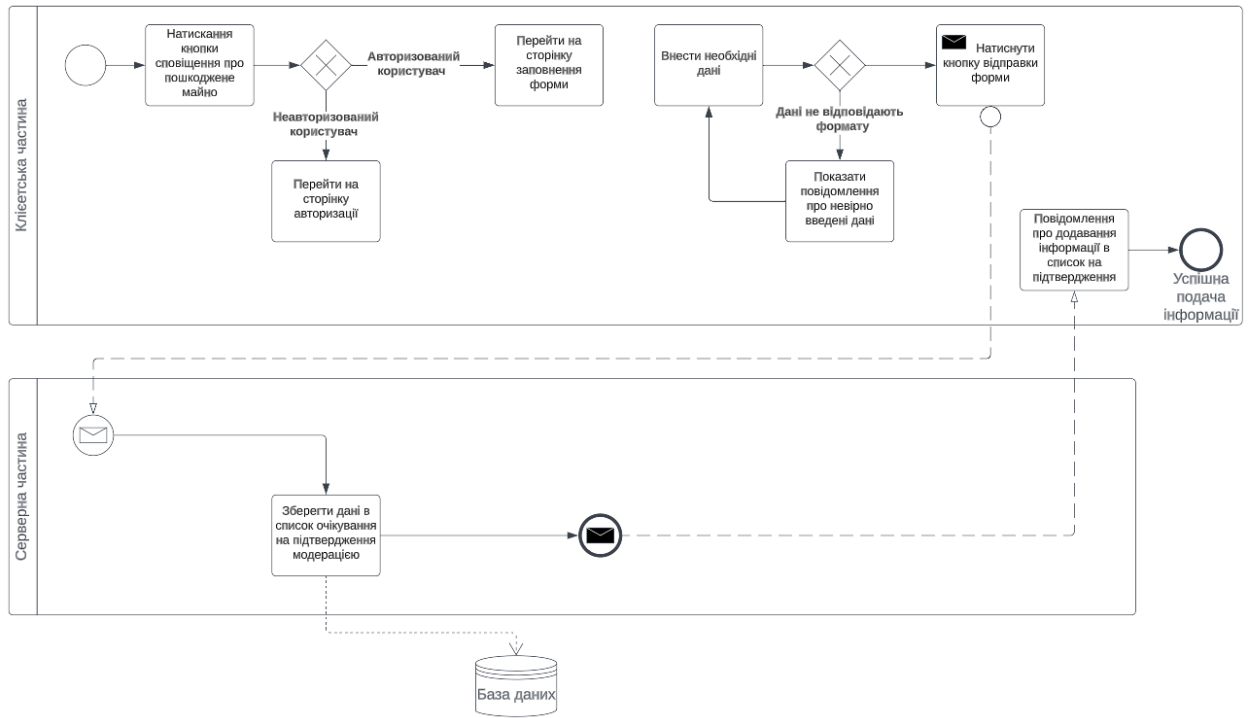


Рисунок 2.5 – Схема бізнес-процесу повідомлення користувачем про зруйноване чи пошкоджене майно

Послідовний опис розгляду заявки про зруйноване або пошкоджене майно менеджером:

- менеджер натискає кнопку перегляду списку заявок;
- обирає заявку, яку планує підтвердити або відхилити;
- натискає одну з доступних кнопок – кнопку відхилення або кнопку підтвердження;
- відбувається спроба збереження даних про голос модератора в базі даних;
- якщо обрана менеджером заявка, яку він відхиляє чи підтверджує, не існує в базі даних – менеджеру буде відображено помилку про те, що обраної заявки не існує;
- в разі існування заявки і успішного збереження відправленого менеджером голосу, він побачить повідомлення про успішну операцію.

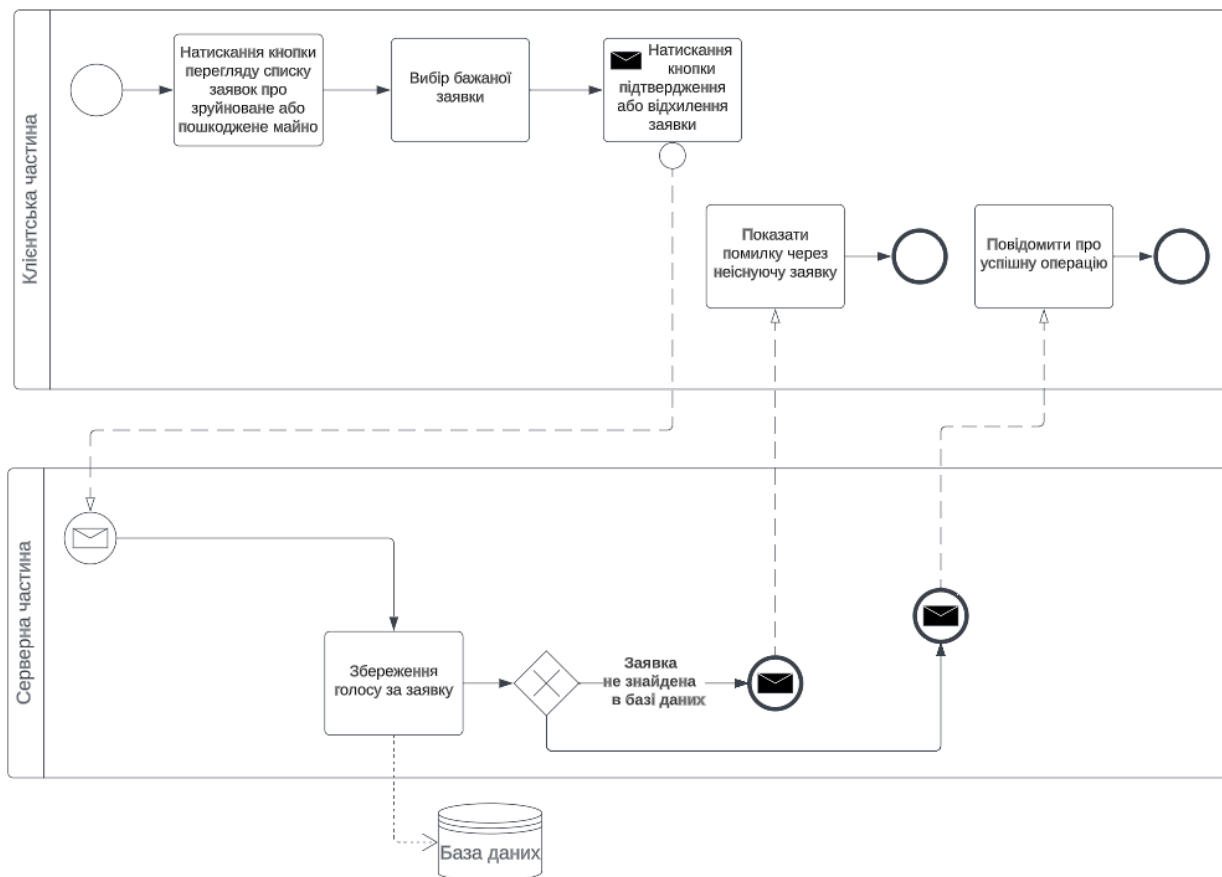


Рисунок 2.6 – Схема бізнес-процесу обробки заявки про пошкоджене чи знищене майно модератором

2.2 Архітектура програмного забезпечення

Враховуючи вимоги, аналіз та моделювання програмного забезпечення, було визначено, що для реалізації проекту будуть використовуватись описані далі технології і методи побудови програмного забезпечення.

Програмне забезпечення є веб-застосунком, який базується на принципі клієнт-серверної архітектури.

Для серверної частини було вибрано платформу .Net Core 3.1 та мову програмування C# версії 7.

Серверна частина застосунку використовує архітектурний шаблон MVC (Model-View-Controller). Цей шаблон допомагає відокремити різні частини системи для забезпечення їх незалежності та легкості змін.

Модель даних відповідає за збереження та обробку даних, виконання бізнес-логіки та взаємодію з базою даних. Вона відокремлена від інтерфейсу

користувача, що дозволяє змінювати та модифікувати модель даних без впливу на вигляд.

Вигляд (інтерфейс користувача) відповідає за представлення даних користувачу та обробку його взаємодії з системою. Він отримує дані з моделі та відображає їх користувачу, а також передає користувачеві введені дані до контролера.

Контролер відповідає за обробку взаємодії користувача з системою. Він приймає введені дані від вигляду, виконує необхідні операції з моделлю даних та повертає оновлені дані відображення для відображення користувачу.

Загальна ідея шаблону MVC полягає у відокремленні різних аспектів системи, щоб зміни в одній частині не впливали на інші частини. Це спрощує розробку, підтримку та розширення системи. [2]

Важливо зазначити, що для успішної реалізації даної архітектури рекомендується використовувати підходи, такі як ін'єкція залежностей (Dependency Injection). Цей шаблон розробки передбачає, що класи залежать від абстракцій (інтерфейсів), а не від конкретних реалізацій, що зменшує залежність між різними компонентами програми. Більшість залежностей визначаються та передаються за допомогою вбудованого контейнера в інструментарії ASP.NET Core – IoC (Inversion of Control). [3]

Цей підхід дозволяє досягти більшої гнучкості та розширюваності системи, оскільки компоненти можуть бути легко замінені або розширені без необхідності вносити зміни на інших рівнях. Використання IoC контейнера спрощує управління залежностями та автоматизує впровадження залежних об'єктів, що полегшує роботу зі змінними компонентами системи.

Для авторизації та аутентифікації буде використано ASP.NET Core Identity. Це вбудований механізм, який використовує файли cookies для ідентифікації авторизованих користувачів. Він забезпечує безпеку та контроль доступу до ресурсів застосунку і надає зручні інструменти для управління користувачами.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 33 |

2.3 Конструювання програмного забезпечення

Для реалізації застосунку будуть використані наступні ключові компоненти та технології:

- База даних: застосунок буде використовувати базу даних SQL Server для зберігання і управління інформацією про пошкоджене майно. Робота з базою даних здійснювалась з допомогою підходу Code-First у поєднанні з Entity Framework Core.

- Серверна частина буде побудована з використанням фреймворку ASP.Net Core MVC, який заснований на архітектурному шаблоні MVC (Model View Controller).

- Для створення інтерфейсу користувача буде використана веб-розробка, зокрема HTML, CSS, JavaScript, бібліотека шаблонів Bootstrap.

- Карта буде відображена з допомогою SVG (Scalable Vector Graphics) картинок, мови розмітки HTML, CSS. [4]

Оригінальні алгоритми та модифікації існуючих:

- алгоритм голосування за подію: менеджери підтверджують чи відхиляють повідомлення від користувачів про пошкодження майна. Якщо повідомлення набере достатньо голосів «за», воно буде додано на карту;

- алгоритм пошуку подій за критерієм: дозволяє здійснювати пошук подій за одним або декількома критеріями;

- алгоритм пошуку подій за ключовими словами: дає можливість знаходити події, в описі яких присутні введені користувачем ключові слова.

База даних містить п'ять основних таблиць: таблиця користувачів, подій, джерел, модераторів та таблиці, що веде облік голосів менеджерів за певну подію. ER діаграма представлена на рисунку 2.7, а схема бази даних на рисунку 2.8.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 34 |

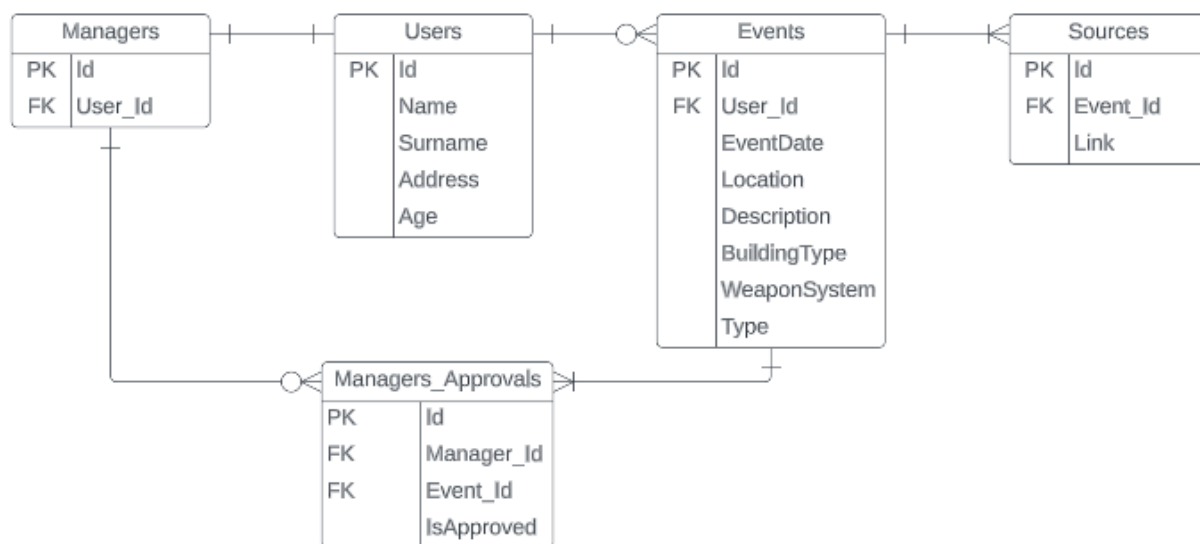


Рисунок 2.7 – ER діаграма сутностей

Таблиця 2.1 – Опис полів таблиці Events

| Назва поля | Тип | Опис |
|--------------|-----------|--|
| Id | Текстовий | Унікальний ідентифікатор. |
| User_Id | Текстовий | Ідентифікатор користувача, який сповістив про подію. |
| EventDate | Дата | Дата події. |
| Location | Текстовий | Місце, до подія трапилась. |
| Description | Текстовий | Детальний опис події |
| BuildingType | Текстовий | Тип будівлі, яка зазнала руйнувань. |
| WeaponSystem | Текстовий | Тип зброї, якою було здійснено обстріл. |
| Type | Текстовий | Визначає, чи подія знаходиться на |

| | | |
|--|--|-----------------------------------|
| | | розгляді, чи вже додана на карту. |
|--|--|-----------------------------------|

Таблиця 2.2 – Опис полів таблиці Sources

| Назва поля | Тип | Опис |
|------------|-----------|--|
| Id | Текстовий | Унікальний ідентифікатор. |
| Link | Текстовий | Посилання на джерело інформації про подію. |
| Event_Id | Текстовий | Ідентифікатор події, про яке сповіщає джерело. |

Таблиця 2.3 – Опис таблиці Users

| Назва поля | Тип | Опис |
|------------|-----------|---------------------------|
| Id | Текстовий | Унікальний ідентифікатор. |
| Name | Текстовий | Ім'я користувача. |
| Surname | Текстовий | Прізвище користувача. |
| Address | Текстовий | Адреса користувача. |
| Age | Числовий | Вік користувача. |

Таблиця 2.4 – Опис таблиці Managers

| Назва поля | Тип | Опис |
|------------|-----------|---------------------------|
| Id | Текстовий | Унікальний ідентифікатор. |

| | | |
|---------|-----------|--|
| User_Id | Текстовий | Ідентифікатор користувача, який сповістив про подію. |
|---------|-----------|--|

Продовження таблиці 2.4

| | | |
|------------|---------|--|
| IsApproved | Булевий | Позитивне чи негативне рішення щодо додавання події. |
|------------|---------|--|

Таблиця 2.5 – Опис таблиці Managers_Approvals

| Назва поля | Тип | Опис |
|------------|-----------|--|
| Id | Текстовий | Унікальний ідентифікатор. |
| Manager_Id | Текстовий | Ідентифікатор менеджера, який проголосував за підтвердження чи відхилення події. |
| Event_Id | Текстовий | Ідентифікатор події, за яку голосував менеджер. |

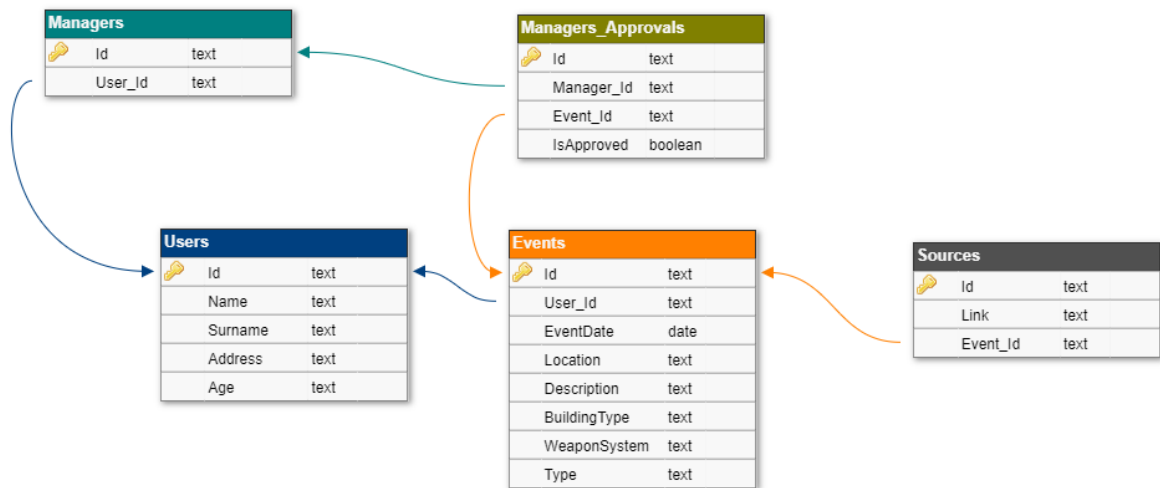


Рисунок 2.8 – Схема бази даних

Об'єктно-орієнтований підхід, використаний при проектуванні програмного забезпечення передбачає організацію системи на основі об'єктів, які взаємодіють між собою для досягнення поставлених цілей. Цей підхід покладається на принципи спадкування, поліморфізму, інкапсуляції та абстракції, що дозволяє розбити систему на окремі модулі та забезпечити їх незалежність та повторне використання.

Можна виділити такі компоненти програми:

- доменні класи (ті, що описують предметні область);
- репозиторії;
- сервіси;
- контролери;
- допоміжні класи.

Таблиця 2.6 – Опис класів, що відображають сутності об'єктної моделі

| Назва класу | Сутність предметної області. |
|-------------|----------------------------------|
| User | Представляє профіль користувача. |

| | |
|---------|--|
| Event | Представляє подію, яка подана користувачем на розгляд або додана на карту. |
| Source | Описує джерела інформації про подію. |
| Manager | Представляє сутність модератора. Наслідник користувача. |

Таблиця 2.7 – Опис класів-репозиторіїв

| Назва класу | Функції |
|-------------------|---|
| UserRepository | <ul style="list-style-type: none"> - Пошук користувача за ідентифікатором. - Додавання нового користувача. - Оновлення користувача - Видалення. |
| ManagerRepository | <ul style="list-style-type: none"> - Пошук менеджера за ідентифікатором. - Пошук менеджера за оціненими подіями. - Додавання. - Видалення. |
| EventRepository | <ul style="list-style-type: none"> - Пошук події за датою, місцем, типом використаної зброї, |

| | |
|------------------|---|
| | <p>типом пошкодженої будівлі, по опису, по типу події (на розгляді чи додана).</p> <ul style="list-style-type: none"> - Додавання події. - Оновлення. - Видалення. |
| SourceRepository | <ul style="list-style-type: none"> - Пошук за зовнішнім ключем події. - Додавання. - Оновлення. - Видалення. |

Продовження таблиці 2.7

| | |
|---------------------|---|
| ApprovalsRepository | <ul style="list-style-type: none"> - Пошук за зовнішнім ключем менеджера або ключем події. - Додавання. - Оновлення. - Видалення. |
|---------------------|---|

Таблиця 2.8 – Опис класів-контролерів

| Назва класу | Функції |
|---------------|--|
| MapController | <ul style="list-style-type: none"> - Організація роботи з картою. - Обробка пошукових запитів. - Навігація. |

| | |
|-------------------|---|
| ManagerController | <ul style="list-style-type: none"> - Організація роботи зі списком подій, поданих користувачами. - Ініціація обробки підтверджень чи відхилень подій менеджерами. - Видалення подій зі списку на підтвердження менеджером. |
| EventController | <ul style="list-style-type: none"> - Організація створення подій. - Ініціація редагування подій. |
| UserController | <ul style="list-style-type: none"> - Ініціація редагування профілю користувача. |

Таблиця 2.9 – Опис сервісів

| Назва | Функції |
|----------------|--|
| UserService | <ul style="list-style-type: none"> - Пошук користувача за ідентифікатором. - Редагування користувача. - Видалення користувача. |
| ManagerService | <ul style="list-style-type: none"> - Обробка підтверджень чи відхилень подій менеджерами. |
| EventService | <ul style="list-style-type: none"> - Редагування подій. - Створення подій. - Видалення подій. - Обробка підтверджених і відхилених менеджером подій. |

| | |
|--|---|
| | - Пошук подій за параметрами і ключовими словами. |
|--|---|

Таблиця 2.9 – Опис допоміжних класів

| Назва | Функції |
|-----------------|---|
| QueryHandler | - Фільтрація подій за параметрами. |
| EventParameters | - Зберігання пошукового запиту користувача при пошуку за параметрами. |

Опис утиліт, бібліотек та іншого стороннього програмного забезпечення, що використовується у розробці наведено в таблиці 2.6.

Таблиця 2.10 – Опис утиліт, бібліотек та іншого стороннього програмного забезпечення

| № п/п | Назва утиліти | Опис застосування |
|-------|---------------------------------|--|
| 1 | JetBrains Rider | Головне середовище розробки програмного забезпечення серверної частини даної роботи. |
| 2 | SQL Server Management Studio 19 | Програмне забезпечення яке надає легкий графічний інтерфейс для доступу до бази даних. |
| 3 | ASP.NET Core 3.1 | Платформа для розробки програмного забезпечення мовою C#. |

| | | |
|---|-----------------------|--|
| 4 | Entity Framework Core | ORM(Object-Relational Mapping) бібліотека, що спрощує роботу з базами даних, надаючи рівень абстракції між даними та програмним кодом. |
| 5 | SQL Server Express | Реляційна система управління базами даних, розроблена компанією Microsoft. |

2.4 Аналіз безпеки даних

У рамках даного застосунку розробляється програмне забезпечення, яке забезпечує захист даних від несанкціонованого доступу за допомогою механізмів авторизації та аутентифікації. Під час входу в систему користувач має ввести логін та пароль, що йому відповідає. При успішній аутентифікації, в клієнтську частину застосунку зберігаються файли cookies, які використовуються для подальших запитів, які цього потребують.

В той час, як доступ до інформації для ознайомлення, а також її пошук за певними критеріями авторизації користувача не вимагають, такі операції, як повідомлення про пошкоджене майно, обробка цих повідомлень менеджерами чи редагування профілю відбуваються при умові аутентифікації користувача.

Заради запобігання несанкціонованого доступу до системи, паролі не зберігаються у відкритому вигляді в базі даних. Змість цього, використовується криптографічна функція для отримання хешів паролів, які потім зберігаються в базі даних.

В застосунку реалізується модель з правами доступу, що дозволяє розділити дані і доступ до них між ролями. Запити, що вимагають авторизації, перевіряються. У разі спроби виконати операцію над даними, до яких він не має прав доступу, система відхиляє запит та запобігає виконанню несанкціонованої операції.

Для аналізу якості програмного забезпечення було використано вбудований інструмент «Code Metrics» в Visual Studio 2022. Результати оцінки показані на рисунку 2.9.

| Hierarchy | Maintainability Index | Cyclomatic Complexity | Depth of Inheritance | Class Coupling | Lines of Sou... | Lines of Ex... |
|---|-----------------------|-----------------------|----------------------|----------------|-----------------|----------------|
| [-] Destruction Map (Debug) | 77 | 521 | 6 | 225 | 5,794 | 1,603 |
| [-] Program | 32 | 8 | 1 | 49 | 115 | 92 |
| [-] AspNetCore | 76 | 342 | 4 | 70 | 3,876 | 1,005 |
| [-] Destruction_Map.Areas.Identity.Data | 78 | 8 | 3 | 4 | 24 | 18 |
| [-] Destruction_Map.Areas.Identity.Pages.Account | 73 | 52 | 2 | 55 | 344 | 125 |
| [-] Destruction_Map.Areas.Identity.Pages.Account.Manage | 83 | 30 | 2 | 14 | 143 | 32 |
| [-] Destruction_Map.Controllers | 75 | 24 | 3 | 39 | 250 | 76 |
| [-] Destruction_Map.Data | 98 | 2 | 6 | 5 | 17 | 1 |
| [-] Destruction_Map.Models | 94 | 35 | 3 | 11 | 104 | 4 |
| [-] Destruction_Map.Models.Middleware | 82 | 15 | 1 | 12 | 109 | 22 |
| [-] Destruction_Map.Views.User | 100 | 1 | 2 | 1 | 9 | 0 |
| [-] DestructionMap.Migrations | 35 | 4 | 2 | 33 | 803 | 228 |
| [-] DestructionMapDAL (Debug) | 97 | 132 | 2 | 44 | 454 | 71 |
| [-] DestructionMapDAL | 96 | 34 | 2 | 24 | 117 | 14 |
| [-] DestructionMapDAL.Entities | 97 | 46 | 1 | 13 | 83 | 26 |
| [-] DestructionMapDAL.Entities.Enums | 100 | 3 | 1 | 0 | 39 | 0 |
| [-] DestructionMapDAL.IRepositories | 100 | 19 | 0 | 13 | 63 | 0 |
| [-] DestructionMapDAL.Repositories | 93 | 30 | 2 | 28 | 152 | 31 |
| [-] DestructionMapLogic (Debug) | 86 | 157 | 2 | 52 | 940 | 193 |
| [-] DestructionMapModel.Abstraction.IMappers | 100 | 2 | 0 | 1 | 12 | 0 |
| [-] DestructionMapModel.Abstraction.IServices | 100 | 19 | 0 | 9 | 55 | 0 |
| [-] DestructionMapModel.Implementation.Mappers | 84 | 8 | 1 | 20 | 121 | 11 |
| [-] DestructionMapModel.Implementation.Services | 67 | 48 | 1 | 49 | 445 | 134 |
| [-] DestructionMapModel.Implementation.Services.QueryProcessing | 75 | 23 | 2 | 13 | 165 | 28 |
| [-] DestructionMapModel.Implementation.Services.Utility | 100 | 8 | 1 | 4 | 12 | 0 |
| [-] DestructionMapModel.Models | 92 | 46 | 1 | 8 | 90 | 20 |
| [-] DestructionMapModel.Models.Enums | 100 | 3 | 1 | 0 | 40 | 0 |

Рисунок 2.9 – Результат перевірки якості програмного забезпечення

Частина застосунку, що відповідає за доступ до даних має високу оцінку обслуговуваності, що означає легкість підтримки, внесення змін чи розширення. Логіка застосунку має відносно невелику цикломатичну складність, що є позитивним фактором, так як виконання великої кількості циклів може негативно впливати на продуктивність програмного забезпечення.

Висновки до розділу

В розділі було показано процеси реєстрації та авторизації користувачів, повідомлення про пошкоджене майно, взаємодію менеджерів з подіями для підтвердження чи відхилення форм користувачів, а також редагування профілю користувача.

Був здійснений опис системи, яка базується на об'єктно орієнтованому підході, який дозволяє розділити її на модулі та забезпечує повторне використання. Компоненти програми, такі як доменні класи, репозиторії, сервіси, контролери та допоміжні класи взаємодіють між собою, утворюючи добре структуровану систему.

Наведено список використаних технологій, бібліотек, утиліт та іншого програмного забезпечення, використаного в застосунку.

Була приділена увага безпеці, яка досягається за допомогою механізмів авторизації та аутентифікації, а також шифруванню паролів, що зберігаються в базі даних.

В цілому, розглянута система є функціональною, забезпечує потрібний рівень безпеки та зручної взаємодії з користувачами. Її проектна реалізація базується на сучасних практиках та технологіях, що дозволяють забезпечити ефективну роботу та задоволення потреб користувачів.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 45 |

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Тестування є, незаперечно, важливим етапом в процесі розробки програмного забезпечення, оскільки воно сприяє попередньому виявленню можливих помилок та недоліків у його функціональності. Цей етап є критичним для забезпечення якості продукту та прямо впливає на задоволення користувачів.

3.2 Опис процесів тестування

Тестування має на меті перевірку програмного продукту згідно з встановленими функціональними вимогами та задля виявлення потенційних помилок, що можуть виникнути під час його роботи.

У процесі тестування програмного продукту використовується комбінація мануального тестування та використання інструменту Postman для автоматизації деяких тестових сценаріїв. Мануальне тестування дозволяє детально перевірити функціональні вимоги програмного забезпечення та виявити потенційні проблеми, які можуть виникнути в реальних умовах використання.

Для ефективного виконання мануальних тестів було виділено список функціональності, яка повинна бути перевірена. Під час виконання тестів були використані різні тестові дані, щоб перевірити функціональність у різних ситуаціях.

Можливі сценарії використання були протестовані. Результати тестів було задокументовано та представлено у відповідних таблицях:

- реєстрація користувача (таблиця 3.1);
- авторизація користувача (таблиця 3.2);
- пошук подій за параметром «час» (таблиця 3.3);
- пошук подій за параметром «тип зброї» (таблиця 3.4);

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 46 |

- пошук подій за параметром «тип враженої будівлі» (таблиця 3.5);
- пошук подій за ключовими словами (таблиця 3.6);
- редагування даних в особистому кабінеті (таблиця 3.7);
- відправка повідомлення про зруйноване або пошкоджене майно (таблиця 3.8);
- обробка отриманого повідомлення про зруйноване або пошкоджене майно менеджером (таблиця 3.9, таблиця 3.10).

Таблиця 3.1 – Тестовий прецедент “Реєстрація користувача”

| Назва | Реєстрація користувача |
|----------------------|---|
| Передумови | Незарєєстрований користувач. |
| Кроки виконання | <ul style="list-style-type: none"> - Перейти до головної сторінки застосунку. - Натиснути на іконку особистого кабінету. - Натиснути кнопку реєстрації на сторінці авторизації. - Ввести необхідні дані(пошта, ім’я, адреса, вік, пароль, підтвердження паролю). - Натиснути кнопку відправки форми. |
| Очікуваний результат | Успішна реєстрація. Перенаправлення на головну сторінку. |
| Отриманий результат | Успішна реєстрація. Перенаправлення на головну сторінку. |

Таблиця 3.2 – Тестовий прецедент “Авторизація користувача”

| | |
|----------------------|--|
| Назва | Авторизація користувача |
| Передумови | Неавторизований користувач |
| Кроки виконання | <ul style="list-style-type: none"> - Перейти до головної сторінки застосунку. - Натиснути на іконку особистого кабінету. - Ввести необхідні дані(пошта, пароль). - Натиснути кнопку відправки форми. |
| Очікуваний результат | Успішна авторизація. Перенаправлення на головну сторінку. |
| Отриманий результат | Успішна реєстрація. Перенаправлення на головну сторінку. |

Таблиця 3.3 – Тестовий прецедент «Пошук подій за параметром “час”»

| | |
|-----------------|--|
| Назва | Пошук подій за параметром «час» |
| Передумови | |
| Кроки виконання | <ul style="list-style-type: none"> - Перейти до головної сторінки застосунку. - Відкрити меню вибору параметрів пошуку. - Обрати бажану дату. - Натиснути кнопку пошуку. |

Продовження таблиці 3.3

| | |
|----------------------|---|
| Очікуваний результат | Замість меню вибору параметрів з'являється список подій, які відповідають обраній даті. |
| Отриманий результат | Замість меню вибору параметрів з'являється список подій, які відповідають обраній даті. |

Таблиця 3.4 – Тестовий прецедент «Пошук подій за параметром «тип зброї»»

| | |
|----------------------|--|
| Назва | Пошук подій за параметром «тип зброї» |
| Передумови | |
| Кроки виконання | <ul style="list-style-type: none"> - Перейти до головної сторінки застосунку. - Відкрити меню вибору параметрів пошуку. - Обрати тип зброї, якою здійснено обстріл. - Натиснути кнопку пошуку. |
| Очікуваний результат | Замість меню вибору параметрів з'являється список подій, які відповідають обраному типу зброї. |
| Отриманий результат | Замість меню вибору параметрів з'являється список подій, які відповідають обраному типу зброї. |

Таблиця 3.5 – Тестовий прецедент “Пошук подій за параметром «тип враженої будівлі»”

| | |
|----------------------|---|
| Назва | Пошук подій за параметром «тип враженої будівлі» |
| Передумови | |
| Кроки виконання | <ul style="list-style-type: none"> - Перейти до головної сторінки застосунку. - Відкрити меню вибору параметрів пошуку. - Обрати тип пошкодженої внаслідок обстрілу будівлі. - Натиснути кнопку пошуку. |
| Очікуваний результат | Замість меню вибору параметрів з'являється список подій, які відповідають обраному типу пошкодженої будівлі. |
| Отриманий результат | Замість меню вибору параметрів з'являється список подій, які відповідають обраному типу пошкодженої будівлі. |

Таблиця 3.6 – Тестовий прецедент “Пошук подій за ключовими словами”

| | |
|------------|----------------------------------|
| Назва | Пошук подій за ключовими словами |
| Передумови | |

Продовження таблиці 3.6

| | |
|----------------------|--|
| Кроки виконання | <ul style="list-style-type: none"> - Перейти до головної сторінки застосунку. - Ввести ключові слова для пошуку у відповідне поле. - Натиснути кнопку пошуку. |
| Очікуваний результат | Список подій оновлюється і містить дані, які відповідають введеному запиту. |
| Отриманий результат | Список подій оновлюється і містить дані, які відповідають введеному запиту. |

Таблиця 3.7 – Тестовий прецедент “Редагування даних в особистому кабінеті”

| | |
|----------------------|---|
| Назва | Редагування даних в особистому кабінеті користувача |
| Передумови | Авторизований користувач |
| Кроки виконання | <ul style="list-style-type: none"> - Натиснути на кнопку особистого кабінету. - Вписати бажані зміни до персональної інформації у відповідні поля. - Натиснути кнопку збереження змін. |
| Очікуваний результат | Особиста інформація в кабінеті користувача оновлюється, відповідно до введених змін. |

Продовження таблиці 3.7

| | |
|---------------------|--|
| Отриманий результат | Особиста інформація в кабінеті користувача оновлюється, відповідно до введених змін. |
|---------------------|--|

Таблиця 3.8 – Тестовий прецедент “Відправка повідомлення про пошкоджене або зруйноване майно”

| | |
|----------------------|---|
| Назва | Відправка повідомлення про зруйноване або пошкоджене майно |
| Передумови | Авторизований користувач |
| Кроки виконання | <ul style="list-style-type: none"> - Натиснути на кнопку додавання події. - На сторінці додавання інформації про подію вписати необхідні дані (місце події, опис, дату, тип використаної зброї, тип враженої будівлі, посилання на ресурси з повідомленням про цю подію). - Натиснути кнопку відправки повідомлення. |
| Очікуваний результат | Подія зберігається в базі даних як така, що потребує підтвердження. Перенаправлення на головну сторінку. |
| Отриманий результат | Подія зберігається в базі даних як така, що потребує підтвердження. Перенаправлення на головну сторінку. |

Таблиця 3.9 – Тестовий прецедент “Підтвердження отриманого повідомлення про зруйноване чи пошкоджене майно менеджером”

| | |
|----------------------|--|
| Назва | Підтвердження отриманого повідомлення про зруйноване чи пошкоджене майно менеджером. |
| Передумови | Авторизований користувач з роллю «менеджер». |
| Кроки виконання | <ul style="list-style-type: none"> - Натиснути на кнопку перегляду подій, які потребують підтвердження. - Натиснути кнопку підтвердження біля бажаної події. |
| Очікуваний результат | В базі даних зберігається підтвердження події менеджером. Подія зникає зі списку вже подій на підтвердження у цього менеджера. |
| Отриманий результат | В базі даних зберігається підтвердження події менеджером. Подія зникає зі списку вже подій на підтвердження у цього менеджера. |

Таблиця 3.10 – Тестовий прецедент “Відхилення отриманого повідомлення про зруйноване чи пошкоджене майно менеджером”

| | |
|------------|---|
| Назва | Відхилення отриманого повідомлення про зруйноване чи пошкоджене майно менеджером. |
| Передумови | Авторизований користувач з роллю «менеджер». |

Продовження таблиці 3.10

| | |
|----------------------|---|
| Кроки виконання | <ul style="list-style-type: none"> - Натиснути на кнопку перегляду подій, які потребують підтвердження. - Натиснути кнопку відхилення біля бажаної події. |
| Очікуваний результат | <p>В базі даних зберігається відхилення події менеджером.</p> <p>Подія зникає зі списку вже подій на підтвердження у цього менеджера.</p> |
| Отриманий результат | <p>В базі даних зберігається відхилення події менеджером.</p> <p>Подія зникає зі списку вже подій на підтвердження у цього менеджера.</p> |

3.3 Опис контрольного прикладу

Контрольний приклад включає в себе взаємодію користувача с веб-застосунком для отримання необхідної інформації. Наведені нижче кроки описують процес, який користувач проходить, щоб отримати бажану інформацію про випадки обстрілів.

1. Авторизація на сайті: неавторизований користувач натискає кнопку особистого кабінету і автоматично перенаправляється на сторінку авторизації. Якщо користувач зареєстрований, він вводить свої дані і натискає кнопку авторизацію (рисунок 3.1).

Log in

| | |
|--|---------------|
| Email | mail@mail.com |
| Password | |
| <input checked="" type="checkbox"/> Remember me? | |
| Register as a new user | |
| <input type="button" value="Log in"/> | |

Рисунок 3.1 – Внесення даних для авторизації

Якщо ж ні – натискає на кнопку реєстрації і вводить свої дані там, як продемонстровано на рисунку 3.2.

Register

| | |
|---|---------------|
| Name | Joe |
| Surname | Biden |
| Address | Street 123 |
| Age | 80 |
| Email | mail@mail.com |
| Password | |
| Confirm Password | |
| <input type="button" value="Register"/> | |

Рисунок 3.2 – Внесення даних для реєстрації

Якщо користувач бажає лише переглядати інформацію про обстріли, то авторизація не обов'язкова.

2. Вибір способу пошуку інформації: користувач може передивитись інформацію по області, або обрати бажані параметри в меню параметрів. Для перегляду інформації про обстріли в конкретній області – користувач натискає на обрану область на карті (рисунок 3.3).

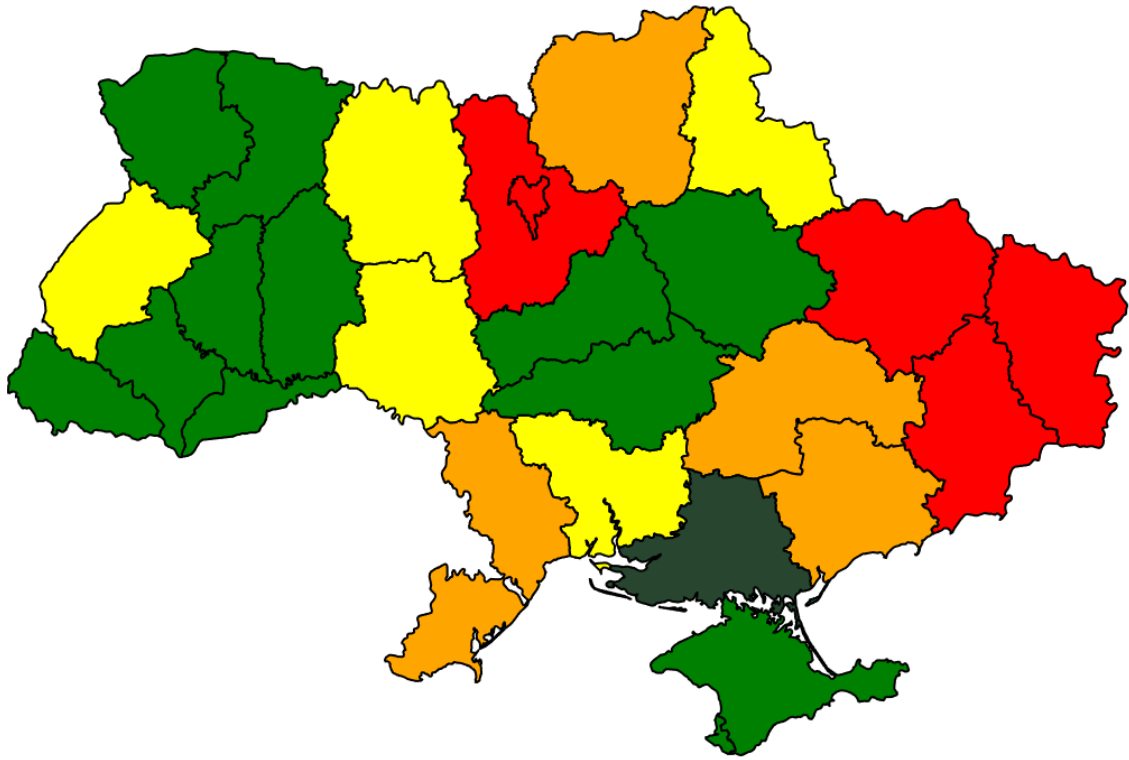
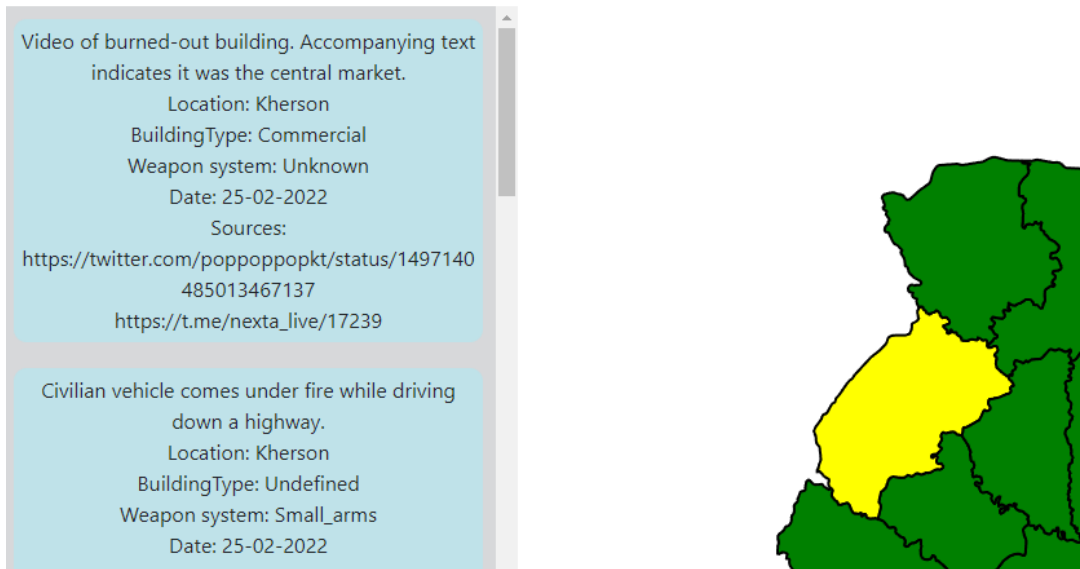


Рисунок 3.3 – Вибір області користувачем

Список подій в обраній області користувач побачить в лівій частині екрану, як на рисунку 3.4.





Video of burned-out building. Accompanying text indicates it was the central market.
Location: Kherson
BuildingType: Commercial
Weapon system: Unknown
Date: 25-02-2022
Sources:
<https://twitter.com/poppoppopkt/status/1497140485013467137>
https://t.me/nexta_live/17239

Civilian vehicle comes under fire while driving down a highway.
Location: Kherson
BuildingType: Undefined
Weapon system: Small_arms
Date: 25-02-2022

Рисунок 3.4 – Результат пошуку по вибраній області

Також користувач може здійснити пошук по обраним параметрам, як представлено на рисунку 3.5.


Destruction Map >  


Select building type:

- Residential
- Industrial
- Administrative
- Healthcare
- School or childcare
- Military
- Commercial
- Religious
- Cultural
- Transport
- Humanitarian
- Food Infrastructure

Select weapon system:

- HE rocket artillery
- Cluster munition
- Incendiary munition
- Cruise missile
- Ballistic missile
- Thermobaric munition
- Vehicle mounted weapon
- Small arms
- Land mines
- Anti air missile
- Loitering munition

Date: 



© 2023 - Destruction Map - [Privacy](#)

Рисунок 3.5 – Процес вибору параметрів пошуку

Результат пошуку буде відображено в тому ж місці, в якому раніше відображалась інформація по обстрілам в обраній області. Це показано на рисунку 3.6.

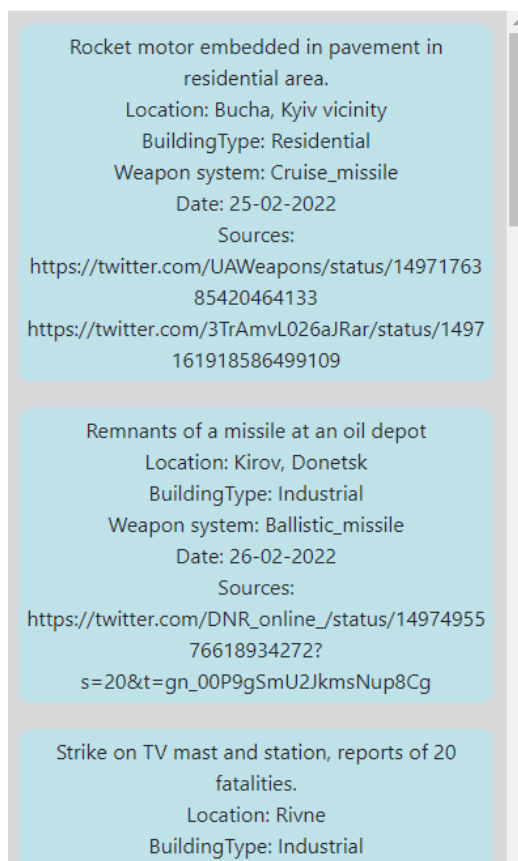


Рисунок 3.6 – Результат пошуку подій по обраним параметрам

Висновки до розділу

В даному розділі було проведено аналіз якості розробленого веб-застосунку. Можливі сценарії використання були послідовно перевірені, а результати описані в таблицях 3.1-3.10. Кожен тест був детально описаний, включаючи його назву, передумови, кроки виконання, очікуваний результат та стан програмного забезпечення після здійснення тестувань.

Також було описано контрольний приклад використання веб-застосунку, який ілюструє процес пошуку інформації користувачем системи.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Для розгортання даного веб-застосунку локально, необхідно мати наступні компоненти.

- Встановити JetBrains Rider або Visual Studio 2019 з необхідними для запуску компонентами, а саме: .Net Core SDK 3.1.
- Встановити СКБД SQL Server 2022 Express.
- Встановити SQL Server Management Studio 19.
- Завантажити проект програмного застосунку.
- Відкрити проект (DestructionMap.sln) в IDE.
- Вказати проект DestructionMap як проект для запуску.
- Вказати власний рядок підключення до бази даних у файлі appSettings.json.
- Запустити команду `dotnet ef database update --project "DestructionMapDAL" --context "DestructionMapContext"` для ініціалізації бази даних.
- Запустити проект.
- Перейти в браузері на сторінку застосунку за шляхом <https://localhost:44333/>.

4.2 Робота з програмним забезпеченням

Інструкція з експлуатації програмного застосунку наведена в керівництві користувача.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 60 |

Висновки до розділу

В цьому розділі проведено опис процесу розгортання веб-застосунку на локальній машині та створення схеми бази даних з наявного в застосунку файлу міграції. Також описано необхідне для роботи програмне забезпечення.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 61 |

ВИСНОВКИ

В результаті виконання цього дипломного проекту, було створено веб-застосунок для моніторингу пошкодженого майна в Україні. Застосунок надає можливість користувачам оцінювати ситуацію в областях, робити висновки на основі кількості обстрілів різних областей. Може здійснюватися пошук як подій як по ключовим словам, так і по параметрам, таким як: час події, місце події, опис події, використана для обстрілу зброя та тип пошкодженої в результаті обстрілу будівлі. Пошук може відбуватись як по кожному окремому параметру, так і по їх комбінації. Присутня можливість сповіщення про випадки обстрілів користувачами і система їх обробки менеджерами застосунку на основі голосування.

Дана робота є значущою через її актуальність, через можливості, які вона може надати як зацікавленим громадянам, так і оперативним службам, благодійним фондам, волонтерам, так і просто для документації випадків обстрілу.

У розділі «Аналіз вимог до програмного забезпечення» було проведено детальний аналіз предметної області та вивчено існуючі програмні продукти, що відносяться до схожого функціонального спектру і їх порівняння з даною роботою. На основі дослідження виявлено функціональні та нефункціональні вимоги до розроблюваного веб застосунку. Крім того, було представлено варіанти використання та сформульовано задачі, які ставляться перед розробкою проекту.

У розділі «Моделювання та конструювання програмного забезпечення» розглянуто використання застосунку як неавторизованими користувачами, так і авторизованими користувачами і менеджерами за допомогою BPMN діаграм. В розділі присутній опис архітектури програмного забезпечення, наведено інформацію про використані технології.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 62 |

У розділі «Аналіз якості та тестування програмного забезпечення» проведено тестування функцій застосунку на відповідність очікуваному сценарію поведінки.

У розділі «Впровадження та супровід програмного забезпечення» було наведено опис розгортання застосунку локально.

Розроблене програмне забезпечення успішно відповідає визначених у попередніх розділах вимогам. Застосунок є зручним у використанні, а якість коду програмного продукту знаходиться на належному рівні.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 63 |

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) [1] Наслідки війни в Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://www.justsecurity.org/81387> .
- 2) [2] Керівництво по ASP.NET Core MVC [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/sharp/aspnetmvc/> .
- 3) [3] ASP.NET Core - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/ASP.NET_Core.
- 4) [4] SVG Map [Електронний ресурс] – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/how-to-make-clickable-svg-map-html-css/>.
- 5) Unit Of Work - Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Unit_Of_Work.
- 6) ASP.NET Identity [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-7.0&tabs=visual-studio>.
- 7) Chain of Responsibility [Електронний ресурс] – Режим доступу до ресурсу: <https://refactoring.guru/design-patterns/chain-of-responsibility>.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.02.81 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 64 |

ДОДАТКИ

Додаток А Звіт подібності



Ім'я користувача:
Лісовиченко Олег Іванович

Дата перевірки:
05.06.2023 07:16:19 EEST

Дата звіту:
05.06.2023 07:22:28 EEST

ID перевірки:
1015420899

Тип перевірки:
Doc vs Internet + Library

ID користувача:
76913

Назва документа: IT-93_Ісмаїлов_ПЗ

Кількість сторінок: 55 Кількість слів: 6463 Кількість символів: 49402 Розмір файлу: 3.07 MB ID файлу: 1015083387

170 слів позначені як "вилучені" та не враховуються у підрахунку слів

12.9% Схожість

Найбільша схожість: 4.24% з джерелом з Бібліотеки (ID файлу: 1015067698)

3.62% Джерела з Інтернету

148

Сторінка 57

12.3% Джерела з Бібліотеки

275

Сторінка 59

1.48% Цитат

Цитати

1

Сторінка 60

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

Немає вилучених Інтернет-джерел

0% Вилученого тексту з Бібліотеки

7

Сторінка 60

| | | | | |
|-------|------|----------|-------|-------|
| | | | | |
| Змін. | Арк. | № докум. | Підп. | Дата. |

КПІ.ІТ-9312.045440.02.81

Арк.

65

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

Веб-застосунок для моніторингу пошкодженого майна в Україні

Текст програми

КПІ.ІТ-9312.045440.03.12

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Ірина Вітковська

Нормоконтроль:

_____ Максим Головченко

Виконавець:

_____ Расім Ісмаїлов

Київ – 2023

Файл Event.cs

```
using DestructionMapModel.Models.Enums;
using Type = DestructionMapModel.Models.Enums.Type;

namespace DestructionMapModel.Models;

public class Event
{
    public string Id { get; set; }
    public string User_Id { get; set; }
    public DateTime EventDate { get; set; }
    public string Location { get; set; }
    public string Description { get; set; }
    public BuildingType BuildingType { get; set; }
    public WeaponSystem WeaponSystem { get; set; }
    public Type Type { get; set; }

    public List<Source> SourceList { get; set; }

    public Event(string userId, DateTime eventDate, string location, string
description, BuildingType buildingType,
    WeaponSystem weaponSystem)
    {
        Random rnd = new Random();
        string id = Guid.NewGuid().ToString(); // "CIV" + rnd.Next(3000,
10000);
        Id = id;
        User_Id = userId;
        EventDate = eventDate;
        Location = location;
        Description = description;
        BuildingType = buildingType;
        WeaponSystem = weaponSystem;
        SourceList = new List<Source>();
    }

    public Event() {}
}
}
```

Файл Manager.cs

```
namespace DestructionMapModel.Models;

public class Manager
{
    public string Id { get; set; }
    public string User_Id { get; set; }

    public Manager(string userId)
    {
        Id = Guid.NewGuid().ToString();
        User_Id = userId;
    }

    public Manager() {}
}
```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 2 |

```
}
```

Файл Source.cs

```
namespace DestructionMapModel.Models;

public class Source
{
    public string Id { get; set; }
    public string Link { get; set; }
    public string Event_Id { get; set; }

    public Source(string link, string eventId)
    {
        Id = Guid.NewGuid().ToString();
        Link = link;
        Event_Id = eventId;
    }

    public Source() {}
}
```

Файл User.cs

```
namespace DestructionMapModel.Models;

public class User
{
    public string Id { get; set; }
    public string Name { get; set; }
    public string Surname { get; set; }
    public string Address { get; set; }
    public int Age { get; set; }

    public User(string name, string surname, string address, int age)
    {
        Id = Guid.NewGuid().ToString();
        Name = name;
        Surname = surname;
        Address = address;
        Age = age;
    }

    public User() {}
}
```

Файл BuildingType.cs

```
namespace DestructionMapModel.Models.Enums;

public enum BuildingType
```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 3 |

```
{
  Residential,
  Industrial,
  Administrative,
  Healthcare,
  School_or_childcare,
  Military,
  Undefined,
  Commercial,
  Religious,
  Cultural,
  Transport,
  Humanitarian,
  Food_Infrastructure
}
```

Файл WeaponSystem.cs

```
namespace DestructionMapModel.Models.Enums;

public enum WeaponSystem
{
  Unknown,
  HE_rocket_artillery,
  Cluster_munition,
  Incendiary_munition,
  Cruise_missile,
  Ballistic_missile,
  Thermobaric_munition,
  Vehicle_mounted_weapon,
  Small_arms,
  Land_mines,
  Anti_air_missile,
  Loitering_munition
}
```

Файл Type.cs

```
namespace DestructionMapModel.Models.Enums;

public enum Type
{
  Event,
  Proposal
}
```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 4 |

Файл IMapper.cs

```
namespace DestructionMapModel.Abstraction.IMappers;

public interface IMapper<TEntity, TModel> : IBackMapper<TEntity, TModel>
{
    TModel ToModel(TEntity entity);
}
}
```

Файл IBackMapper.cs

```
namespace DestructionMapModel.Abstraction.IMappers;

public interface IBackMapper<TEntity, TModel>
{
    TEntity ToEntity(TModel model);
}
}
```

Файл EventMapper.cs

```
using DestructionMapDAL.Entities;
using DestructionMapDAL.Entities.Enums;
using DestructionMapModel.Abstraction.IMappers;
using DestructionMapModel.Models;
using Type = DestructionMapDAL.Entities.Enums.Type;

namespace DestructionMapModel.Implementation.Mappers;

public class EventMapper : IMapper<EventEntity, Event>
{
    private IMapper<SourceEntity, Source> _sourceMapper = new SourceMapper();

    /*
    public EventMapper(IMapper<SourceEntity, Source> sourceMapper)
    {
        _sourceMapper = sourceMapper;
    }
    */

    public EventEntity ToEntity(Event model)
    {
        return new EventEntity
        {
            Id = model.Id,
            User_Id = model.User_Id,
            Description = model.Description,
            Location = model.Location,
            WeaponSystem = (WeaponSystem)model.WeaponSystem,
            BuildingType = (BuildingType)model.BuildingType,
            Type = (Type)model.Type,
            EventDate = model.EventDate,
        }
    }
}
```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 5 |

```

        SourceList = new List<SourceEntity>(model.SourceList.ConvertAll(s
=> _sourceMapper.ToEntity(s)))
    };
}

public Event ToModel(EventEntity entity)
{
    return new Event
    {
        Id = entity.Id,
        User_Id = entity.User_Id,
        Description = entity.Description,
        Location = entity.Location,
        WeaponSystem = (Models.Enums.WeaponSystem)entity.WeaponSystem,
        BuildingType = (Models.Enums.BuildingType)entity.BuildingType,
        Type = (Models.Enums.Type)entity.Type,
        EventDate = entity.EventDate,
        SourceList = new List<Source>(entity.SourceList.ConvertAll(s =>
_sourceMapper.ToModel(s)))
    };
}
}
}

```

Файл ManagerMapper.cs

```

using DestructionMapDAL.Entities;
using DestructionMapModel.Abstraction.IMappers;
using DestructionMapModel.Models;

namespace DestructionMapModel.Implementation.Mappers;

public class ManagerMapper : IMapper<ManagerEntity, Manager>
{
    public ManagerEntity ToEntity(Manager model)
    {
        return new ManagerEntity
        {
            Id = model.Id,
            User_Id = model.User_Id
        };
    }

    public Manager ToModel(ManagerEntity entity)
    {
        return new Manager
        {
            Id = entity.Id,
            User_Id = entity.User_Id
        };
    }
}

```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
|-------|------|----------|-------|-------|

Файл SourceMapper.cs

```
using DestructionMapDAL.Entities;
using DestructionMapModel.Abstraction.IMappers;
using DestructionMapModel.Models;

namespace DestructionMapModel.Implementation.Mappers;

public class SourceMapper : IMapper<SourceEntity, Source>
{
    public SourceEntity ToEntity(Source model)
    {
        return new SourceEntity
        {
            Id = model.Id,
            Event_Id = model.Event_Id,
            Link = model.Link
        };
    }

    public Source ToModel(SourceEntity entity)
    {
        return new Source
        {
            Id = entity.Id,
            Event_Id = entity.Event_Id,
            Link = entity.Link
        };
    }
}
```

Файл UserMapper.cs

```
using DestructionMapDAL.Entities;
using DestructionMapModel.Abstraction.IMappers;
using DestructionMapModel.Models;

namespace DestructionMapModel.Implementation.Mappers;

public class UserMapper : IMapper<UserEntity, User>
{
    public UserEntity ToEntity(User model)
    {
        return new UserEntity
        {
            Id = model.Id,
            Address = model.Address,
            Age = model.Age,
            Name = model.Name,
            Surname = model.Surname
        };
    }

    public User ToModel(UserEntity entity)
    {
        return new User
        {

```

| | | | | |
|-------|------|----------|-------|-------|
| | | | | |
| Змін. | Арк. | № докум. | Підп. | Дата. |

```

        Id = entity.Id,
        Address = entity.Address,
        Age = entity.Age,
        Name = entity.Name,
        Surname = entity.Surname
    };
}
}

```

Файл IEventService.cs

```

using DestructionMapModel.Implementation.Services.Utility;
using DestructionMapModel.Models;
using DestructionMapModel.Models.Enums;

namespace DestructionMapModel.Abstraction.IServices;

public interface IEventService
{
    public List<Event> GetAll();

    public List<Event> GetByLocation(string region);

    public List<Event> FindByKeyWords(string keyWords);

    public List<Event> GetByWeaponSystem(WeaponSystem weaponSystem);

    public List<Event> GetByBuildingType(BuildingType buildingType);

    public List<Event> GetByDate(DateTime eventDate);

    public List<Event> GetByDescription(string description);

    public List<Event> FindByParameters(EventParameters eventParameters);

    public void CreateEvent(string userId, DateTime eventDate, string
location, string description, BuildingType buildingType, WeaponSystem
weaponSystem, string sources);

    public List<Event> GetEventsToApprove();

    public Dictionary<string, int> GetEventsIntensity();
}

```

Файл IManagerService.cs

```

using DestructionMapModel.Models;

namespace DestructionMapModel.Abstraction.IServices;

public interface IManagerService
{

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Вмін. | Арк. | № докум. | Підп. | Дата. | | 8 |

```

    public Manager GetManager (string id);

    public bool VoteForEvent(string eventId, string managerId, bool
isApproved);

    public bool IsAlreadyVoder(string eventId, string managerId);

    public void ApproveOrRemove(string eventId);

    public bool CreateManager(string userId);

}

```

Файл IUserService.cs

```

using DestructionMapModel.Models;

namespace DestructionMapModel.Abstraction.IServices;

public interface IUserService
{
    public bool CreateUser(string id, string name, string surname, int age,
string address);

    public bool UpdateUser(string id, string name, string surname, int age,
string address);

    public User GetUserById(string id);

}

```

Файл EventService.cs

```

using DestructionMapDAL;
using DestructionMapDAL.Entities;
using DestructionMapModel.Abstraction.IMappers;
using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Implementation.Services.QueryProcessing;
using DestructionMapModel.Implementation.Services.Utility;
using DestructionMapModel.Models;
using DestructionMapModel.Models.Enums;
using Microsoft.EntityFrameworkCore;
using Type = DestructionMapModel.Models.Enums.Type;

namespace DestructionMapModel.Implementation.Services;

public class EventService : IEventService
{
    private IUnitOfWork _unitOfWork;
    private IMapper<EventEntity, Event> _eventMapper;
    private IMapper<SourceEntity, Source> _sourceMapper;

    public EventService(IUnitOfWork unitOfWork, IMapper<EventEntity, Event>
eventMapper, IMapper<SourceEntity, Source> sourceMapper)
    {

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 9 |

```

        _unitOfWork = unitOfWork;
        _eventMapper = eventMapper;
        _sourceMapper = sourceMapper;
    }

    public List<Event> GetAll()
    {
        return new
List<Event>(_unitOfWork.Events.GetAll_IncludeAll().ToList().ConvertAll(_event
Mapper.ToModel));
    }

    public List<Event> GetByLocation(string region)
    {
        return new
List<Event>(_unitOfWork.Events.GetByLocation(region.ToLower()).ToList().Conve
rtAll(_eventMapper.ToModel));
    }

    public List<Event> GetByWeaponSystem(WeaponSystem weaponSystem)
    {
        return new List<Event>(_unitOfWork.Events
.GetByWeaponSystem((DestructionMapDAL.Entities.Enums.WeaponSystem) weaponSyste
m).ToList()
        .ConvertAll(_eventMapper.ToModel));
    }

    public List<Event> GetByBuildingType(BuildingType buildingType)
    {
        return new List<Event>(_unitOfWork.Events
.GetByBuildingType((DestructionMapDAL.Entities.Enums.BuildingType)buildingType
).ToList()
        .ConvertAll(_eventMapper.ToModel));
    }

    public List<Event> GetByDate(DateTime eventDate)
    {
        return new List<Event>(_unitOfWork.Events
        .GetByDate(eventDate).ToList()
        .ConvertAll(_eventMapper.ToModel));
    }

    public List<Event> GetByDescription(string description)
    {
        return new List<Event>(_unitOfWork.Events
        .GetByDescription(description).ToList()
        .ConvertAll(_eventMapper.ToModel));
    }

    public List<Event> FindByParameters(EventParameters eventParameters)
    {
        QueryHandler byDate = new DateHandler();
        QueryHandler byWeaponSystem = new WeaponSystemHandler();
        QueryHandler byBuildingType = new BuildingTypeHandler();
        QueryHandler byLocation = new LocationHandler();

        byDate.Successor = byWeaponSystem;
        byWeaponSystem.Successor = byBuildingType;
        byBuildingType.Successor = byLocation;

        List<Event> result = byDate.HandleRequest(eventParameters, this,

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 10 |

```

    GetAll());

    return result;
}

public List<Event> FindByKeyWords(string keyWords)
{
    string[] keyWordSplit = keyWords.Split();

    List<Event> match = new List<Event>();

    List<Event> matchByDescription = new List<Event>();
    List<Event> matchByLocation = new List<Event>();
    List<Event> matchByWeaponSystem = new List<Event>();
    List<Event> matchByBuildingType = new List<Event>();

    WeaponSystem weaponSystem;
    BuildingType buildingType;

    foreach (string word in keyWordSplit)
    {
        matchByDescription = GetByDescription(word);
        matchByLocation = GetByLocation(word);

        if (Enum.TryParse(word, true, out weaponSystem))
            matchByWeaponSystem = GetByWeaponSystem(weaponSystem);

        if (Enum.TryParse(word, true, out buildingType))
            matchByBuildingType = GetByBuildingType(buildingType);
    }

    match.AddRange(matchByDescription);
    match.AddRange(matchByLocation);
    match.AddRange(matchByWeaponSystem);
    match.AddRange(matchByBuildingType);

    return match.Distinct(new EventComparer()).ToList();
}

public void CreateEvent(string userId, DateTime eventDate, string
location, string description, BuildingType buildingType, WeaponSystem
weaponSystem, string sources)
{
    string[] sourcesSplit = sources.Split("\n");

    Event eventToSave = new Event(userId, eventDate, description,
location, buildingType, weaponSystem);
    //eventToSave.Id = "TestEvent"; //REMOVE THIS
    eventToSave.Id = Guid.NewGuid().ToString();
    eventToSave.Type = Type.Proposal;

    List<Source> sourcesForEvent = new List<Source>();
    foreach (var source in sourcesSplit)
    {
        var sourceEntity = new Source(){Id = Guid.NewGuid().ToString(),
Event_Id = eventToSave.Id, Link = source}; // Generate a unique Id for each
Source entity
        sourcesForEvent.Add(sourceEntity);
    }
    eventToSave.SourceList = sourcesForEvent;
}

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.IT-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 11 |

```

        _unitOfWork.Events.Create(_eventMapper.ToEntity(eventToSave));
        _unitOfWork.Save();
    }

    public List<Event> GetEventsToApprove(string userId)
    {
        List<Managers_Approvals> alreadyApprovedByThisManager =
        _unitOfWork.Approvals.Find(a => a.Manager_Id == userId) .ToList();

        List<Event> eventsToApprove =
        _unitOfWork.Events.GetEventsToApprove().ToList().ConvertAll(e =>
        _eventMapper.ToModel(e));

        foreach (var eventToCheck in eventsToApprove)
        {
            foreach (var alreadyApproved in alreadyApprovedByThisManager)
            {
                if (eventToCheck.Id == alreadyApproved.Event_Id)
                {
                    eventsToApprove.Remove(eventToCheck);
                }
            }
        }

        return _unitOfWork.Events.GetEventsToApprove().ToList().ConvertAll(e
=> _eventMapper.ToModel(e));
    }

    public Dictionary<string, int> GetEventsIntensity()
    {
        Dictionary<string, int> eventIntensity = new Dictionary<string,
int>();

        Dictionary<string, string> Id_Location = new Dictionary<string,
string>()
        {
            { "UKR283", "Crimea" },
            { "UKR284", "Mikolaiv" },
            { "UKR285", "Chernihiv" },
            { "UKR286", "Rivne" },
            { "UKR288", "Chernivtsi" },
            { "UKR289", "Ivano-Frankivsk" },
            { "UKR290", "Khmelnyskyi" },
            { "UKR291", "Lviv" },
            { "UKR292", "Ternopil" },
            { "UKR293", "Transcarpathia" },
            { "UKR318", "Volyn" },
            { "UKR319", "Cherkasy" },
            { "UKR320", "Kirovohrad" },
            { "UKR321", "Kyiv" },
            { "UKR322", "Odessa" },
            { "UKR323", "Vinnytsia" },
            { "UKR324", "Zhytomyr" },
            { "UKR325", "Sumy" },
            { "UKR326", "Dnipro" },
            { "UKR327", "Donetsk" },
            { "UKR328", "Kharkiv" },
            { "UKR329", "Luhansk" },
            { "UKR330", "Poltava" },
            { "UKR331", "Zaporizhzhia" },
        }
    }

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 12 |

```

        {"UKR4826", "Kyiv"},
        {"UKR4827", "Kherson"},
        {"UKR5482", "Crimea"}
    };

    foreach (var location in Id_Location)
    {
        eventIntensity.Add(location.Key,
            _unitOfWork.Events.GetIntensityByLocation(location.Value));
    }

    return eventIntensity;
}
}

```

Файл ManagerService.cs

```

using DestructionMapDAL;
using DestructionMapDAL.Entities;
using DestructionMapModel.Abstraction.IMappers;
using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Implementation.Mappers;
using DestructionMapModel.Models;
using Type = DestructionMapDAL.Entities.Enums.Type;

namespace DestructionMapModel.Implementation.Services;

public class ManagerService : IManagerService
{
    private IUnitOfWork _unitOfWork;
    private IMapper<ManagerEntity, Manager> _managerMapper;

    public ManagerService(IUnitOfWork unitOfWork, IMapper<ManagerEntity,
        Manager> managerMapper)
    {
        _unitOfWork = unitOfWork;
        _managerMapper = managerMapper;
    }

    public Manager GetManager(string id)
    {
        ManagerEntity manager = _unitOfWork.Managers.Get(id);

        if (manager != null)
            return _managerMapper.ToModel(manager);
        else
            return null;
    }

    public bool CreateManager(string userId)
    {
        _unitOfWork.Managers.Create(new ManagerEntity{Id = userId, User_Id =
            userId});
        _unitOfWork.Save();
        return true;
    }
}

```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
|-------|------|----------|-------|-------|

```

public bool VoteForEvent(string eventId, string managerId, bool
isApproved)
{
    EventEntity eventToVote = _unitOfWork.Events.Get(eventId);
    ManagerEntity manager = _unitOfWork.Managers.Get(managerId);
    bool isAlreadyVoted = IsAlreadyVoted(eventId, managerId);
    if (eventToVote != null && manager != null && !isAlreadyVoted)
    {
        Managers_Approvals managerVote = new Managers_Approvals()
        { Id = Guid.NewGuid().ToString(), Event_Id = eventId,
Manager_Id = managerId, IsApproved = isApproved };
        _unitOfWork.Approvals.Create(managerVote);
        _unitOfWork.Save();
        ApproveOrRemove(eventId);
        return true;
    }
    else
        return false;
}

public bool IsAlreadyVoted(string eventId, string managerId)
{
    Managers_Approvals? existingVote = _unitOfWork.Approvals.Find(a =>
a.Manager_Id == managerId && a.Event_Id == eventId).FirstOrDefault();
    if (existingVote != null)
        return true;
    else
        return false;
}

public void ApproveOrRemove(string eventId)
{
    int approvingThreshold = 2;

    EventEntity eventToCheck = _unitOfWork.Events.Get(eventId);
    List<Managers_Approvals> eventApprovals =
_unitOfWork.Approvals.GetByEventId(eventId).ToList();

    int approveCount = 0;
    int declineCount = 0;

    foreach (Managers_Approvals managerVote in eventApprovals)
    {
        if (managerVote.IsApproved)
            approveCount++;
        else
            declineCount++;

        if (approveCount >= approvingThreshold)
        {
            eventToCheck.Type = Type.Event;
            _unitOfWork.Events.Update(eventToCheck);
            _unitOfWork.Save();
        }
        else if (declineCount >= approvingThreshold)
        {
            _unitOfWork.Events.Remove(eventToCheck);
            _unitOfWork.Save();
        }
        approveCount = declineCount = 0;
    }
}

```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
|-------|------|----------|-------|-------|

```
}  
  
}
```

Файл UserService.cs

```
using DestructionMapDAL;  
using DestructionMapDAL.Entities;  
using DestructionMapModel.Abstraction.IMappers;  
using DestructionMapModel.Abstraction.IServices;  
using DestructionMapModel.Models;  
using Microsoft.IdentityModel.Tokens;  
  
namespace DestructionMapModel.Implementation.Services;  
  
public class UserService : IUserService  
{  
  
    private IUnitOfWork _unitOfWork;  
    private IMapper<UserEntity, User> _userMapper;  
  
    public UserService(IUnitOfWork unitOfWork, IMapper<UserEntity, User>  
userMapper)  
    {  
        _unitOfWork = unitOfWork;  
        _userMapper = userMapper;  
    }  
  
    public bool CreateUser(string id, string name, string surname, int age,  
string address)  
    {  
        if (_unitOfWork.Users.Find(u => u.Id == id).FirstOrDefault() == null)  
        {  
            _unitOfWork.Users.Create(new UserEntity()  
                { Id = id, Address = address, Age = age, Name = name, Surname  
= surname });  
            _unitOfWork.Save();  
            return true;  
        }  
        return false;  
    }  
  
    public bool UpdateUser(string id, string name, string surname, int age,  
string address)  
    {  
        UserEntity? userToChange = _unitOfWork.Users.Find(u=> u.Id ==  
id).FirstOrDefault();  
        if (userToChange != null)  
        {  
            userToChange.Name = name.IsNullOrEmpty() ? userToChange.Name :  
userToChange.Name = name;  
            userToChange.Surname = surname.IsNullOrEmpty() ?  
userToChange.Surname : userToChange.Surname = surname;  
            userToChange.Age = age == 0 ? userToChange.Age : userToChange.Age  
= age;  
            userToChange.Address = address.IsNullOrEmpty() ?  
userToChange.Address : userToChange.Address = address;  
  
            _unitOfWork.Save();  
        }  
    }  
}
```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
| | | | | |

КПІ.ІТ-9312.045440.03.12

Арк.

15

```

        return true;
    }

    return false;
}

public User GetUserById(string Id)
{
    UserEntity? userEntity = _unitOfWork.Users.Find(u => u.Id ==
Id).FirstOrDefault();
    return _userManager.ToModel(userEntity);
}
}

```

Файл EventParameters.cs

```

using DestructionMapModel.Models.Enums;
using Type = System.Type;

namespace DestructionMapModel.Implementation.Services.Utility;

public class EventParameters
{
    public DateTime EventDate { get; set; }
    public string Location { get; set; }
    public List<BuildingType> BuildingType { get; set; }
    public List<WeaponSystem> WeaponSystem { get; set; }
}

```

Файл QueryHandler.cs

```

using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Implementation.Services.Utility;
using DestructionMapModel.Models;

namespace DestructionMapModel.Implementation.Services.QueryProcessing;

abstract class QueryHandler
{
    public QueryHandler Successor { get; set; }

    public abstract List<Event> HandleRequest(EventParameters
eventParameters, IEventService eventService, List<Event> events);
}

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Вмін. | Арк. | № докум. | Підп. | Дата. | | 16 |

Файл BuildingTypeHandler.cs

```
using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Implementation.Services.Utility;
using DestructionMapModel.Models;
using DestructionMapModel.Models.Enums;

namespace DestructionMapModel.Implementation.Services.QueryProcessing;

class BuildingTypeHandler : QueryHandler
{
    public override List<Event> HandleRequest(EventParameters
eventParameters, IEventService eventService,
    List<Event> events)
    {
        try
        {
            List<Event> byBuildingType = events;

            if (eventParameters.BuildingType.Count != 0)
            {
                List<Event> sumOfSelected = new List<Event>();

                foreach (var building in eventParameters.BuildingType)
                {
                    sumOfSelected.AddRange(eventService.GetByBuildingType(building));
                }

                byBuildingType = byBuildingType.Intersect(sumOfSelected, new
EventComparer()).ToList();
            }

            if (Successor != null)
                return Successor.HandleRequest(eventParameters, eventService,
byBuildingType);
            else
                return byBuildingType;
        }
        catch
        {
            if (Successor != null)
                return Successor.HandleRequest(eventParameters, eventService,
events);
            else
                return events;
        }
    }
}
```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Вмін. | Арк. | № докум. | Підп. | Дата. | | 17 |

Файл DateHandler.cs

```
using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Implementation.Services.Utility;
using DestructionMapModel.Models;

namespace DestructionMapModel.Implementation.Services.QueryProcessing;

class DateHandler : QueryHandler
{
    public override List<Event> HandleRequest(EventParameters
eventParameters, IEventService eventService,
    List<Event> events)
    {
        if (eventParameters.EventDate != DateTime.MinValue)
        {
            List<Event> eventsByDate =
eventService.GetByDate(eventParameters.EventDate).Intersect(events, new
EventComparer()).ToList();

            if (Successor != null)
                return Successor.HandleRequest(eventParameters, eventService,
eventsByDate);
            else
                return eventsByDate;
        }
        else
        {
            if (Successor != null)
                return Successor.HandleRequest(eventParameters, eventService,
events);
            else
                return events;
        }
    }
}
```

Файл EventComparer.cs

```
using DestructionMapModel.Models;

namespace DestructionMapModel.Implementation.Services.QueryProcessing;

public class EventComparer : IEqualityComparer<Event>
{
    public bool Equals(Event x, Event y)
    {
        //TODO - Add null handling
        return x.Id == y.Id;
    }

    public int GetHashCode(Event obj)
    {
        return obj.Id.GetHashCode();
    }
}
```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Вмін. | Арк. | № докум. | Підп. | Дата. | | 18 |

```
}
```

Файл LocationHandler.cs

```
using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Implementation.Services.Utility;
using DestructionMapModel.Models;
using Microsoft.IdentityModel.Tokens;

namespace DestructionMapModel.Implementation.Services.QueryProcessing;

class LocationHandler : QueryHandler
{
    public override List<Event> HandleRequest(EventParameters
eventParameters, IEventService eventService,
    List<Event> events)
    {
        if (!eventParameters.Location.IsNullOrEmpty())
        {
            List<Event> eventsByLocation =
eventService.GetByLocation(eventParameters.Location).Intersect(events, new
EventComparer()).ToList();

            if (Successor != null)
                return Successor.HandleRequest(eventParameters, eventService,
eventsByLocation);
            else
                return eventsByLocation;
        }
        else
        {
            if (Successor != null)
                return Successor.HandleRequest(eventParameters, eventService,
events);
            else
                return events;
        }
    }
}
```

Файл LocationHandler.cs

```
using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Implementation.Services.Utility;
using DestructionMapModel.Models;
using DestructionMapModel.Models.Enums;

namespace DestructionMapModel.Implementation.Services.QueryProcessing;

class WeaponSystemHandler : QueryHandler
{
```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Вмін. | Арк. | № докум. | Підп. | Дата. | | 19 |

```

    public override List<Event> HandleRequest(EventParameters
eventParameters, IEventService eventService,
    List<Event> events)
    {
        try
        {
            List<Event> byWeaponSystem = events;

            if (eventParameters.WeaponSystem.Count != 0)
            {
                List<Event> sumOfSelected = new List<Event>();
                foreach (var weapon in eventParameters.WeaponSystem)
                {
                    sumOfSelected.AddRange(eventService.GetByWeaponSystem(weapon));
                }

                byWeaponSystem = byWeaponSystem.Intersect(sumOfSelected, new
EventComparer()).ToList();
            }

            if (Successor != null)
                return Successor.HandleRequest(eventParameters, eventService,
byWeaponSystem);
            else
                return byWeaponSystem;
        }
        catch
        {
            if (Successor != null)
                return Successor.HandleRequest(eventParameters, eventService,
events);
            else
                return events;
        }
    }
}

```

Файл EventWebModel.cs

```

using DestructionMapModel.Models;
using DestructionMapModel.Models.Enums;
using Type = System.Type;

namespace Destruction_Map.Models;

public class EventWebModel
{
    public string Id { get; set; }

    public string User_Id { get; set; }
    public DateTime EventDate { get; set; }
    public string Location { get; set; }
    public string Description { get; set; }
    public BuildingType BuildingType { get; set; }
    public WeaponSystem WeaponSystem { get; set; }
}

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 20 |

```

    public Type Type { get; set; }

    public List<Source> SourceList { get; set; }
}

```

Файл EventWebModelMapper.cs

```

using DestructionMapModel.Models;

namespace Destruction_Map.Models;

public class EventWebModelMapper
{
    public EventWebModel ToEventWebModel(Event model)
    {
        return new EventWebModel()
        {
            Id = model.Id,
            Description = model.Description,
            BuildingType = model.BuildingType,
            WeaponSystem = model.WeaponSystem,
            Location = model.Location,
            EventDate = model.EventDate,
            User_Id = model.User_Id,
            SourceList = model.SourceList
        };
    }
}

```

Файл User.cs

```

using System.ComponentModel.DataAnnotations.Schema;
using Microsoft.AspNetCore.Identity;

namespace Destruction_Map.Models;

public class User : IdentityUser
{
    [PersonalData]
    [Column]
    public string Name { get; set; }

    [PersonalData]
    [Column]
    public string Surname { get; set; }

    [PersonalData]
    [Column]
    public string Address { get; set; }

    [PersonalData]

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Вмін. | Арк. | № докум. | Підп. | Дата. | | 21 |

```

    [Column]
    public int Age { get; set; }

}

```

Файл Manager.cs

```

using System.ComponentModel.DataAnnotations.Schema;
using Microsoft.AspNetCore.Identity;

namespace Destruction_Map.Models;

public class Manager : IdentityUser
{
    [Column]
    [ForeignKey(nameof(User))]
    public string User_Id { get; set; }
}

```

Файл AddEvent.cshtml

```

@using DestructionMapModel.Models.Enums
@model EventWebModel

@{
    ViewBag.Title = "Add Event";
}

<header>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-
white border-bottom box-shadow mb-3">
        <div class="container-fluid">
            <a class="navbar-brand" asp-area="" asp-controller="Map" asp-
action="Index">Destruction Map</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
                aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="navbar-collapse collapse d-sm-inline-flex justify-
content-end">

                <partial name="_LoginPartial"/>
            </div>
        </div>
    </nav>
</header>

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 22 |

```

<main style="display: flex; justify-content: center" role="main"><br/>
  <div style="justify-content: center; width: 30%" class="addEventMenu">
    <h2 align="center" style="margin-top: 30px">Add event</h2><br/>
    <form method="post" asp-action="AddEvent">
      Event date:<br/> <input class="form-control" required type="date"
name="eventDate"/><br/><br/>
      <input placeholder="Location" class="form-control" required
type="text" name="location"/><br/><br/>
      <div class="invalid-feedback">
        Please provide a valid city.
      </div>
      <input placeholder="Description" class="form-control" required
type="text" name="description"/><br/><br/>
      <select class="buildingTypeSelect form-control" required
name="buildingType">
        <option value="">-- Select building type --</option>
        @foreach (BuildingType building in
Enum.GetValues(typeof(BuildingType)))
        {
          if (building != BuildingType.Undefined)
          {
            <option
value="@building">@building.ToString().Replace("_", " ").Replace(";",
"")</option>
          }
        }
      </select><br/><br/>
      <select class="weaponSystemSelect form-control" required
name="weaponSystem">
        <option value="">-- Select weapon system --</option>
        @foreach (WeaponSystem weapon in
Enum.GetValues(typeof(WeaponSystem)))
        {
          if (weapon != WeaponSystem.Unknown)
          {
            <option
value="@weapon">@weapon.ToString().Replace("_", " ").Replace(";",
"")</option>
          }
        }
      </select><br/><br/>
      <textarea required placeholder="Enter sources. Start each one
from new line" class="form-control" type="text" name="sources"
multiple></textarea><br/><br/>
      <div style="display: flex; justify-content: center">
        <input style="align-items: center" class="btn btn-primary btn-lg"
type="submit" value="Send"/>
      </div>
    </form>
  </div>
</main>

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 23 |


```

                <button asp-action="Approve" class="btn btn-success"
name="eventId" value="@eventToApprove.Id">Approve</button>
                <button asp-action="Decline" class="btn btn-success"
name="eventId" value="@eventToApprove.Id">Decline</button>
            </form>
        </div>
    }
</div>

<style>
    .eventToApproveContainer{
        width: 30%;
        height: 80vh;
        background-color: #ced4da;
        border-radius: 10px;
        overflow: auto;
    }
    .eventToApprove{
        margin: 7px 25px 30px 25px;
        display: block;
        border: 3px solid black;
        border-radius: 20px;
        padding: 10px;
    }

    .eventApprovalLabel{
        white-space: normal;
        word-break: break-word;
        overflow-wrap: break-word;
    }
</style>

</main>

```

Файл Index.cshtml

```

@using Destruction_Map.Controllers
@using Microsoft.AspNetCore.Mvc.TagHelpers
@using Microsoft.JSInterop
@using System.Reflection.Metadata

@{
    ViewData["Title"] = "Destruction Map";
}
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

<header>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-
white border-bottom box-shadow mb-3">
        <div class="container-fluid">
            <a class="navbar-brand" asp-area="" asp-controller="Map" asp-
action="Index">Destruction Map</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
                aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
        </div>
    </nav>

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 25 |

```

        </button>
        <div class="navbar-collapse collapse d-sm-inline-flex justify-
content-between">
            <ul class="navbar-nav flex-grow-1">
                <li class="nav-item">
                    
                </li>
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-
controller="Map" asp-action="Index"></a>
                </li>
                <li class="nav-item">
                    <!--<a class="nav-link text-dark" asp-area="" asp-
controller="Map" asp-action="Privacy"> -->
                    <form method="get" class="searchForm" asp-
action="SearchByKeyWords">
                        <input class="searchForm_textInput" type="text"
name="keyWords" />
                        <button type="submit" class="search_btn"></button>
                    </form>

                </li>

            </ul>

            @if (User.IsInRole("Manager"))
            {
                <a class="nav-link text-dark approveEventLink" asp-
controller="Manager" asp-action="ApproveEvent">Approve event</a>
            }

            <a class="nav-link text-dark addEventLink" asp-area="" asp-
controller="Event" asp-action="AddEvent">Add event</a>

            <partial name="_LoginPartial" />
        </div>
    </div>
</nav>
</header>

<script>

function OnArrowClick()
{
    var infoContainer = document.getElementById("infoContainer");
    if (infoContainer.style.display === "none")
        infoContainer.style.display = "flex";
    else
        infoContainer.style.display = "none";
}

function OpenInfoPanel(clicked_id)
{
    let infoContainer = document.getElementById("infoContainer");
    infoContainer.style.display = "flex";
    console.log(" ----- " + clicked_id);
}

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.IT-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 26 |

```

</script>

<main role="main" >

  <div class="MapContent">

    <div class="infoContainer" id="infoContainer">

      <!--await Html.PartialAsync("InfoBox"); -->

      @if (Model != null)
      {
        foreach (var ev in Model)
        {
          <div class="event">
            @ev.Description<br/>
            Location: @ev.Location<br/>
            BuildingType: @ev.BuildingType<br/>
            Weapon system: @ev.WeaponSystem<br/>
            Date: @ev.EventDate.ToString("dd-MM-yyyy")<br/>
            Sources: <br/>

            @if (ev.SourceList != null && ev.SourceList.Count>0)
            {
              int linkCount = 0;

              @for (int i = 0; i<ev.SourceList.Count; i++)
              {
                if (linkCount < 2 && ev.SourceList[i].Link != "HIDDEN")
                {
                  @ev.SourceList[i].Link; <br/>
                  linkCount++;
                }
              }
            }

          </div>

        }
      }
      else
      {
        Html.RenderPartial("ShowParams");
      }

    </div>

    <div class="map">
      @RenderPartial("_MapPartial"); //Тут <svg></svg> об'єкт мапи
    </div>

    <script>

    function ChangeColorOnHover(item, color){

      var itemToExperiment = document.getElementById(item);
      itemToExperiment.addEventListener('mouseover', function (){
        itemToExperiment.style.fill = "#27452f";
      });
      itemToExperiment.addEventListener('mouseout', function (){
        itemToExperiment.style.fill = color;
      });
    }
  }

```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
| | | | | |

```

    }

</script>
    @foreach (var item in ViewData)
    {
        string color = "";
        int result;
        int.TryParse(item.Value.ToString(), out result);
        if (result < 3)
        {
            color = "green";
        }
        else if (result is >= 3 and <= 10)
        {
            color = "yellow";
        }
        else if (result is > 10 and <= 15)
        {
            color = "orange";
        }
        else if (result > 15)
        {
            color = "red";
        }

        <script>
            var elemToChange =
document.getElementById("@item.Key").style.fill = "@color";
            ChangeColorOnHover("@item.Key", "@color");

        </script>
    }

</div>

</main>

```

Файл InfoBox.cshtml

```

@model List<EventWebModel>

@if (Model != null)
{
    foreach (var ev in Model)
    {
        <div class="event">
            @ev.Description<br/>
            text text text text text text text
            text text text text text text text
            Location: @ev.Location<br/>
            BuildingType: @ev.BuildingType

        </div>
    }
}

```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
|-------|------|----------|-------|-------|

Файл ShowParams.cshtml

```
@using DestructionMapModel.Implementation.Services.Utility
@using DestructionMapModel.Models.Enums
@using static DestructionMapModel.Models.Enums.BuildingType
@using static DestructionMapModel.Models.Enums.WeaponSystem

<style>

    .parameters_container{
    display: flex;
    justify-content: center;
    align-items: center;
    }

</style>

<div class="parameters_container">

    <form asp-action="SearchByParams" method="get">

        <div class="buildingTypeBox">
            <h3>Select building type:</h3>

            @foreach (Enum buildingType in Enum.GetValues(typeof(BuildingType)))
            {
                @if ((BuildingType)buildingType != BuildingType.Undefined)
                {
                    <div class="form-check buildingType_container">
                        <input class="form-check-input" name="selectedBuildings"
type="checkbox" value="@buildingType" id="@buildingType">
                        <label class="form-check-label" for="@buildingType">
                            @buildingType.ToString().Replace("_", " ").Replace(";", " ")
                        </label>
                    </div>
                }
            }
        </div>

        <div class="weaponSystemBox">
            <h3>Select weapon system:</h3>

            @foreach (Enum weaponSystem in Enum.GetValues(typeof(WeaponSystem)))
            {
                @if ((WeaponSystem)weaponSystem != WeaponSystem.Unknown)
                {
                    <div class="form-check weaponSystem_container">
                        <input class="form-check-input" name="selectedWeaponSystems"
type="checkbox" value="@weaponSystem" id="@weaponSystem">
                        <label class="form-check-label" for="@weaponSystem">
                            @weaponSystem.ToString().Replace("_", " ").Replace(";", " ")
                        </label>
                    </div>
                }
            }
        }
    }
</form>
</div>
```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
|-------|------|----------|-------|-------|

```

</div>

<label style="margin-right: 8px">Date: </label><input type="date"
class="form-control" name="date"/>

<input style="margin-bottom: 5px" placeholder="Location" class="form-
control" name="location" type="text"/>

<input class="btn btn-primary btn-lg" type="submit" value="Search"/>

</form>

</div>

```

Файл EventController.cs

```

using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Models.Enums;
using Microsoft.AspNetCore.Mvc;

namespace Destruction_Map.Controllers;

public class EventController : Controller
{
    private readonly ILogger<MapController> _logger;
    private IEventService _eventService;

    public EventController(ILogger<MapController> logger, IEventService
eventService)
    {
        _logger = logger;
        _eventService = eventService;
    }

    public IActionResult AddEvent()
    {
        return View();
    }

    [HttpPost]
    public IActionResult AddEvent(DateTime eventDate, string location, string
description, BuildingType buildingType, WeaponSystem weaponSystem, string
sources)
    {
        string[] sourcesSplit = sources.Split("\n");

        Console.WriteLine(eventDate + " " + location + " " + description + " " +
buildingType + " " + weaponSystem + " " + sourcesSplit.Length);

        _eventService.CreateEvent("TestUser", eventDate, location,
description, buildingType, weaponSystem, sources);

        return View();
    }
}

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 30 |

Файл ManagerController.cs

```
using Destruction_Map.Areas.Identity.Data;
using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Models;
using DestructionMapModel.Models.Enums;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;

namespace Destruction_Map.Controllers;

[Authorize(Policy = "ManagerOnly")] //
public class ManagerController : Controller
{
    private readonly ILogger<MapController> _logger;
    private IEventService _eventService;
    private IUserService _userService;
    private IManagerService _managerService;
    private readonly UserManager<ApplicationUser> _userManager;

    public ManagerController(ILogger<MapController> logger, IEventService
eventService,
        IUserService userService, IManagerService managerService,
        UserManager<ApplicationUser> userManager)
    {
        _logger = logger;
        _eventService = eventService;
        _userService = userService;
        _managerService = managerService;
        _userManager = userManager;
    }

    public IActionResult ApproveEvent()
    {
        string userId = _userManager.GetUserId(User);
        List<Event> eventsToApprove =
_eventService.GetEventsToApprove(userId);

        return View(eventsToApprove);
    }

    [HttpPost]
    public IActionResult Approve(string eventId)
    {
        string userId = _userManager.GetUserId(User);

        _managerService.VoteForEvent(eventId, userId, true);

        return RedirectToAction("ApproveEvent");
    }

    [HttpPost]
    public IActionResult Decline(string eventId)
    {

```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
|-------|------|----------|-------|-------|

КПІ.ІТ-9312.045440.03.12

Арк.

31

```

        string userId = _userManager.GetUserId(User);
        _managerService.VoteForEvent(eventId, userId, false);

        return RedirectToAction("ApproveEvent");
    }
}

```

Файл MapController.cs

```

using Destruction_Map.Areas.Identity.Data;
using Microsoft.AspNetCore.Mvc;
using Destruction_Map.Models;
using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Implementation.Services.Utility;
using DestructionMapModel.Models;
using DestructionMapModel.Models.Enums;
using Microsoft.AspNetCore.Identity;

namespace Destruction_Map.Controllers;

public class MapController : Controller
{
    private readonly ILogger<MapController> _logger;
    private IEventService _eventService;
    private EventWebModelMapper _eventWebModelMapper = new
EventWebModelMapper();
    private readonly UserManager<ApplicationUser> _userManager;

    public MapController(ILogger<MapController> logger, IEventService
eventService, UserManager<ApplicationUser> userManager)
    {
        _logger = logger;
        _eventService = eventService;
        _userManager = userManager;
    }

    public IActionResult Index()
    {
        ViewBag.UserId = _userManager.GetUserId(User); //Only if logged

        ColorMap();
        return View();
    }

    [NonAction]
    public void ColorMap()
    {
        Dictionary<string, int> eventsIntensity =
_eventService.GetEventsIntensity();

        foreach (var location in eventsIntensity)
        {
            ViewData[location.Key] = location.Value;
        }
    }
}

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 32 |

```

        Console.WriteLine(location.Key + " ---- " + location.Value);
    }
}

public IActionResult Privacy()
{
    //_eventService.CreateEvent();
    return View();
}

public IActionResult InfoBox()
{
    return PartialView();
}

[HttpGet]
public IActionResult SearchByKeyWords(string keyWords)
{
    List<Event> filteredList = _eventService.FindByKeyWords(keyWords);
    ViewBag.Events = filteredList;

    ColorMap();
    return View("Index", filteredList);
}

public IActionResult GetEvents(string location, int page = 1)
{
    List<EventWebModel> list =
        _eventService.GetByLocation(location).ConvertAll(e=>
            _eventWebModelMapper.ToEventWebModel(e));

    ColorMap();

    return View("Index", list);
    //return RedirectToAction("Index", list);
}

[HttpGet]
public IActionResult SearchByParams(List<BuildingType> selectedBuildings,
List<WeaponSystem> selectedWeaponSystems, string location, DateTime date)
{
    EventParameters eventParameters = new EventParameters()
    {
        Location = location, EventDate = date, BuildingType =
selectedBuildings,
        WeaponSystem = selectedWeaponSystems
    };

    List<EventWebModel> list =
        _eventService.FindByParameters(eventParameters).ConvertAll(e =>
            _eventWebModelMapper.ToEventWebModel(e));

    foreach (var building in selectedBuildings)
    {
        //Console.WriteLine(building.ToString());
    }

    ColorMap();
    return View("Index", list);
}

```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
|-------|------|----------|-------|-------|

```

public IActionResult ShowParams ()
{
    //ViewBag.ShowParams = true;
    EventWebModel eventModel = new EventWebModel () { Id = "3rf234f34" };
    return PartialView(eventModel);
}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None,
NoStore = true)]
public IActionResult Error ()
{
    return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
}
}

```

Файл UserController.cs

```

using Destruction_Map.Areas.Identity.Data;
using DestructionMapModel.Abstraction.IServices;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;

namespace Destruction_Map.Controllers;

[Authorize]
public class UserController : Controller
{
    private readonly ILogger<MapController> _logger;
    private IUserService _userService;
    private readonly UserManager<ApplicationUser> _userManager;

    public UserController(ILogger<MapController> logger, IUserService
userService)
    {
        _logger = logger;
        _userService = userService;
    }

    [HttpPost]
    public IActionResult UpdateUser(string userId, string name, string
surname, string address, int age)
    {
        _userService.UpdateUser(userId, name, surname, age, address);

        return RedirectToAction("");
    }

    public IActionResult UserPage ()
    {
        string id = _userManager.GetUserId(User);
        _userService.GetUserById(id);
        return View();
    }
}

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Вмін. | Арк. | № докум. | Підп. | Дата. | | 34 |

```
}
```

Файл Program.cs

```
using Destruction_Map.Areas.Identity.Data;
using Microsoft.AspNetCore.Identity;
using Microsoft.EntityFrameworkCore;
using Destruction_Map.Models;
using Destruction_Map.Models.Middleware;
using DestructionMapDAL;
using DestructionMapDAL.Entities;
using DestructionMapModel.Abstraction.IMappers;
using DestructionMapModel.Abstraction.IServices;
using DestructionMapModel.Implementation.Mappers;
using DestructionMapModel.Implementation.Services;
using DestructionMapModel.Models;
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.AspNetCore.Authorization;
using Microsoft.EntityFrameworkCore.Query;
using Microsoft.Extensions.Options;
using Microsoft.Extensions.Primitives;
using Microsoft.IdentityModel.Tokens;
using Microsoft.VisualBasic;
using Manager = DestructionMapModel.Models.Manager;
using User = Destruction_Map.Models.User; //DestructionMapModel.Models.User;
using Destruction_Map.Data;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
var connectionString =
builder.Configuration.GetConnectionString("DefaultConnection") ??
    throw new InvalidOperationException("Connection string
'DefaultConnection' not found.");

var authConnectionString =
builder.Configuration.GetConnectionString("AuthDbContextConnection") ??
    throw new InvalidOperationException("Connection string
'AuthDbContext' not found.");

builder.Services.AddDbContext<DestructionMapContext>(options =>
    options
        .UseLazyLoadingProxies()
        .UseSqlServer(connectionString));

builder.Services.AddDbContext<AuthDbContext>(options =>
    options.UseSqlServer(authConnectionString));

builder.Services.AddDefaultIdentity<ApplicationUser>(options =>
    options.SignIn.RequireConfirmedAccount = false)
    .AddRoles<IdentityRole>()
    .AddEntityFrameworkStores<AuthDbContext>();

builder.Services.Configure<IdentityOptions>(options =>
{
    options.Password.RequireUppercase = false;
    options.Password.RequiredLength = 4;
    options.Password.RequireLowercase = false;
```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 35 |

```

        options.Password.RequireNonAlphanumeric = false;
    });

builder.Services.AddDatabaseDeveloperPageExceptionFilter();
builder.Services.AddRazorPages();

builder.Services.AddControllersWithViews();

builder.Services.AddAuthentication();

builder.Services.AddAuthorization(options =>
{
    options.AddPolicy("ManagerOnly", policy =>
policy.RequireRole("Manager"));
    options.AddPolicy("UserOnly", policy => policy.RequireRole("User"));
});

builder.Services.AddTransient<DestructionMapContext>();
builder.Services.AddTransient<IUnitOfWork, UnitOfWork>();
builder.Services.AddTransient<IMapper<SourceEntity, Source>, SourceMapper>();
builder.Services.AddTransient<IMapper<UserEntity,
DestructionMapModel.Models.User>, UserMapper>();
builder.Services.AddTransient<IMapper<ManagerEntity, Manager>,
ManagerMapper>();
builder.Services.AddTransient<IMapper<EventEntity, Event>, EventMapper>();
builder.Services.AddTransient<IEventService, EventService>();
builder.Services.AddTransient<IUserService, UserService>();
builder.Services.AddTransient<IManagerService, ManagerService>();

var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.UseMigrationsEndPoint();
}
else
{
    app.UseExceptionHandler("/Home/Error");

    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseDefaultFiles();

app.UseRouting();

app.UseAuthentication();
app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Map}/{action=Index}/{id?}");
app.MapRazorPages();

app.Run();

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 36 |

Файл EventEntity.cs

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using DestructionMapDAL.Entities.Enums;
using Type = DestructionMapDAL.Entities.Enums.Type;

namespace DestructionMapDAL.Entities;

[Table("Events")]
public class EventEntity
{
    public string Id { get; set; }

    [ForeignKey(nameof(UserEntity))]
    public string User_Id { get; set; }

    public DateTime EventDate { get; set; }

    public string Location { get; set; }

    public string Description { get; set; }

    [EnumDataType(typeof(BuildingType))]
    public BuildingType BuildingType { get; set; }

    [EnumDataType(typeof(WeaponSystem))]
    public WeaponSystem WeaponSystem { get; set; }

    public virtual List<SourceEntity> SourceList { get; set; }

    [EnumDataType(typeof(Type))]
    public Type Type { get; set; }
}
```

Файл ManagerEntity.cs

```
using System.ComponentModel.DataAnnotations.Schema;

namespace DestructionMapDAL.Entities;

[Table("Managers")]
public class ManagerEntity
{
    public string Id { get; set; }

    [ForeignKey(nameof(UserEntity))]
    public string User_Id { get; set; }
}
```

Файл Managers_Approval.cs

```
using System.ComponentModel.DataAnnotations.Schema;

namespace DestructionMapDAL.Entities;

[Table("Managers_Approvals")]
public class Managers_Approvals
{
    public string Id { get; set; }

    [ForeignKey(nameof(ManagerEntity))]
    public string Manager_Id { get; set; }

    [ForeignKey(nameof(EventEntity))]
    public string Event_Id { get; set; }

    public bool IsApproved { get; set; }
}
```

Файл SourceEntity.cs

```
using System.ComponentModel.DataAnnotations.Schema;

namespace DestructionMapDAL.Entities;

[Table("Sources")]
public class SourceEntity
{
    public string Id { get; set; }

    public string Link { get; set; }

    [ForeignKey(nameof(EventEntity))]
    public string Event_Id { get; set; }
}
```

Файл UserEntity.cs

```
using System.ComponentModel.DataAnnotations.Schema;

namespace DestructionMapDAL.Entities;

[Table("Users")]
public class UserEntity
{
    public string Id { get; set; }

    public string Name { get; set; }
}
```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 38 |

```

public string Surname { get; set; }

public string Address { get; set; }

public int Age { get; set; }

}

```

Файл IEventRepository.cs

```

using DestructionMapDAL.Entities;
using DestructionMapDAL.Entities.Enums;
using Type = DestructionMapDAL.Entities.Enums.Type;

namespace DestructionMapDAL.IRepositories;

public interface IEventRepository : IRepository<EventEntity, string>
{
    public IEnumerable<EventEntity> GetByWeaponSystem(WeaponSystem
weaponSystem);

    public IEnumerable<EventEntity> GetByBuildingType(BuildingType
buildingType);

    public IEnumerable<EventEntity> GetByType(Type type);

    public IEnumerable<EventEntity> GetAll_IncludeAll();

    public IEnumerable<EventEntity> GetByLocation(string location);

    public IEnumerable<EventEntity> GetByDate(DateTime eventDate);

    public IEnumerable<EventEntity> GetByDescription(string description);

    public IEnumerable<EventEntity> GetEventsToApprove();

    public int GetIntensityByLocation(string location);
}

```

Файл IRepository.cs

```

using System.Linq.Expressions;

namespace DestructionMapDAL.IRepositories;

public interface IRepository<TEntity, TIdentity> where TEntity : class
{
    IEnumerable<TEntity> GetAll();

    TEntity Get(TIdentity Id);
}

```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 39 |

```

        IEnumerable<TEntity> Find(Expression<Func<TEntity, bool>> predicate);

        void Create(TEntity entity);

        void Update(TEntity entity);

        void Remove(TEntity entity);
    }

```

Файл EventRepository.cs

```

using DestructionMapDAL.Entities;
using DestructionMapDAL.Entities.Enums;
using DestructionMapDAL.IRepositories;
using Microsoft.EntityFrameworkCore;
using Type = DestructionMapDAL.Entities.Enums.Type;

namespace DestructionMapDAL.Repositories;

public class EventRepository : Repository<EventEntity, string>,
    IEventRepository
{
    public EventRepository(DestructionMapContext context) : base(context)
    {
    }

    public IEnumerable<EventEntity> GetByWeaponSystem(WeaponSystem
weaponSystem)
    {
        return db.Events.Where(e => e.WeaponSystem == weaponSystem);
    }

    public IEnumerable<EventEntity> GetByBuildingType(BuildingType
buildingType)
    {
        return db.Events.Where(e => e.BuildingType == buildingType);
    }

    public IEnumerable<EventEntity> GetByType(Type type)
    {
        return db.Events.Where(e => e.Type == type);
    }

    public IEnumerable<EventEntity> GetAll_IncludeAll()
    {
        return db.Events.Include(e => e.SourceList);
    }

    public IEnumerable<EventEntity> GetByLocation(string location)
    {
        return db.Events.Where(e => e.Location.ToLower().Contains(location)
||
e.Description.ToLower().Contains(location)
        .Include(e => e.SourceList);
    }
}

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 40 |

```

public IEnumerable<EventEntity> GetByDate(DateTime eventDate)
{
    return db.Events.Where(e => e.EventDate == eventDate);
}

public IEnumerable<EventEntity> GetByDescription(string description)
{
    return db.Events.Where(e => e.Description == description);
}

public IEnumerable<EventEntity> GetEventsToApprove()
{
    return db.Events.Where(e => e.Type == Type.Proposal).Include(e =>
e.SourceList);
}

public int GetIntensityByLocation(string location)
{
    return db.Events.Count(e =>
e.Location.ToLower().Contains(location.ToLower()));
}
}

```

Файл Repository.cs

```

using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Linq.Expressions;
using DestructionMapDAL.IRepositories;
using EntityState = Microsoft.EntityFrameworkCore.EntityState;

namespace DestructionMapDAL.Repositories;

public class Repository<TEntity, TIdentity> : IRepository<TEntity, TIdentity>
where TEntity : class
{
    protected readonly DestructionMapContext db;

    public Repository(DestructionMapContext context)
    {
        db = context;
    }

    public void Create(TEntity entity)
    {
        db.Set<TEntity>().Add(entity);
    }

    public void Remove(TEntity entity)
    {
        db.Set<TEntity>().Remove(entity);
    }
}

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 41 |

```

    public IEnumerable<TEntity> Find(Expression<Func<TEntity, bool>>
predicate)
    {
        return db.Set<TEntity>().Where(predicate);
    }

    public TEntity Get(TIdentity id)
    {
        return db.Set<TEntity>().Find(id);
    }

    public IEnumerable<TEntity> GetAll()
    {
        return db.Set<TEntity>();
    }

    public void Update(TEntity entity)
    {
        db.Entry(entity).State = EntityState.Modified;
    }
}

```

Файл DestructionMapContext.cs

```

using System.Configuration;
using DestructionMapDAL.Entities;
using Microsoft.EntityFrameworkCore;

namespace DestructionMapDAL;

public class DestructionMapContext : DbContext
{
    //private string connectionString;

    public DestructionMapContext(DbContextOptions<DestructionMapContext>
options) : base(options)
    {
        //connectionString = "Default connection";
        Database.EnsureCreated();
    }

    /*
    protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder)
        => optionsBuilder
            .UseLazyLoadingProxies()
            .UseSqlServer(connectionString);
    */

    public virtual DbSet<EventEntity> Events { get; set; }
    public virtual DbSet<UserEntity> Users { get; set; }
    public virtual DbSet<SourceEntity> Sources { get; set; }
    public virtual DbSet<ManagerEntity> Managers { get; set; }
}

```

| | | | | |
|-------|------|----------|-------|-------|
| Змін. | Арк. | № докум. | Підп. | Дата. |
| | | | | |

```
public virtual DbSet<Managers_Approvals> Approvals { get; set; } //
```

Файл IUnitOfWork.cs

```
using DestructionMapDAL.IRepositories;  
namespace DestructionMapDAL;  
public interface IUnitOfWork  
{  
    IEventRepository Events { get; }  
    IUserRepository Users { get; }  
    IManagerRepository Managers { get; }  
    ISourceRepository Sources { get; }  
    IApprovalsRepository Approvals { get; }  
    int Save();  
}
```

Файл UnitOfWork.cs

```
using DestructionMapDAL.IRepositories;  
using DestructionMapDAL.Repositories;  
namespace DestructionMapDAL;  
public class UnitOfWork : IUnitOfWork  
{  
    private readonly DestructionMapContext _context;  
    public IEventRepository Events { get; private set; }  
    public IUserRepository Users { get; private set; }  
    public IManagerRepository Managers { get; private set; }  
    public ISourceRepository Sources { get; private set; }  
    public IApprovalsRepository Approvals { get; private set; }  
    public UnitOfWork(DestructionMapContext context)  
    {  
        _context = context;  
        Events = new EventRepository(_context);  
        Users = new UserRepository(_context);  
        Managers = new ManagerRepository(_context);  
        Sources = new SourceRepository(_context);  
    }  
}
```

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 43 |

```

        Approvals = new ApprovalsRepository(_context);
    }

    public int Save()
    {
        return _context.SaveChanges();
    }

    private bool disposed = false;

    public void Dispose(bool disposing)
    {
        if (!this.disposed)
        {
            if (disposing)
            {
                _context.Dispose();
            }
        }
    }

    public void Dispose()
    {
        Dispose(true);
        GC.SuppressFinalize(this);
    }
}

```

Файл Register.cshtml

```

@page
@model RegisterModel
@{
    ViewData["Title"] = "Register";
}
<header>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-
white border-bottom box-shadow mb-3">
        <div class="container-fluid">
            <a class="navbar-brand" asp-area="" asp-controller="Map" asp-
action="Index">Destruction Map</a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
                aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="navbar-collapse collapse d-sm-inline-flex justify-
content-between">

                <!--partial name="_LoginPartial"/-->
            </div>
        </div>
    </nav>

```

| | | | | | | | |
|-------|------|----------|-------|-------|--|--------------------------|------|
| | | | | | | КПІ.IT-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | | 44 |

```

</header>

<main role="main" >

    <div style="margin-top: 5%" class="row justify-content-center">

        <h1 align="center" >@ViewData["Title"]</h1>

        <div class="col-md-4">
            <form id="registerForm" asp-route-returnUrl="@Model.ReturnUrl"
method="post">

                <hr/>
                <div asp-validation-summary="ModelOnly" class="text-danger"
role="alert"></div>
                <div class="form-floating mb3">
                    <input asp-for="Input.Name" class="form-control"
autocomplete="name" aria-required="true" placeholder="Name"/>
                    <label asp-for="Input.Name">Name</label>
                    <span asp-validation-for="Input.Name" class="text-
danger"></span>
                </div>
                <div class="form-floating mb3">
                    <input asp-for="Input.Surname" class="form-control"
autocomplete="surname" aria-required="true" placeholder="Surname"/>
                    <label asp-for="Input.Surname">Surname</label>
                    <span asp-validation-for="Input.Surname" class="text-
danger"></span>
                </div>
                <div class="form-floating mb3">
                    <input asp-for="Input.Address" class="form-control"
autocomplete="address" aria-required="true" placeholder="Address"/>
                    <label asp-for="Input.Address">Address</label>
                    <span asp-validation-for="Input.Address" class="text-
danger"></span>
                </div>
                <div class="form-floating mb3">
                    <input asp-for="Input.Age" class="form-control"
autocomplete="age" aria-required="true" placeholder="Age"/>
                    <label asp-for="Input.Age">Age</label>
                    <span asp-validation-for="Input.Age" class="text-
danger"></span>
                </div>
                <div class="form-floating mb-3">
                    <input asp-for="Input.Email" class="form-control"
autocomplete="username" aria-required="true" placeholder="name@example.com"/>
                    <label asp-for="Input.Email">Email</label>
                    <span asp-validation-for="Input.Email" class="text-
danger"></span>
                </div>
                <div class="form-floating mb-3">
                    <input asp-for="Input.Password" class="form-control"
autocomplete="new-password" aria-required="true" placeholder="password"/>
                    <label asp-for="Input.Password">Password</label>
                    <span asp-validation-for="Input.Password" class="text-
danger"></span>
                </div>
                <div class="form-floating mb-3">
                    <input asp-for="Input.ConfirmPassword" class="form-
control" autocomplete="new-password" aria-required="true"
placeholder="password"/>

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 45 |

```

        <label asp-for="Input.ConfirmPassword">Confirm
Password</label>
        <span asp-validation-for="Input.ConfirmPassword"
class="text-danger"></span>
    </div>
    <button id="registerSubmit" type="submit" class="w-100 btn
btn-lg btn-primary">Register</button>
    </form>
</div>

</div>

</main>

@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}

```

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.03.12 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 46 |

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

Веб-застосунок для моніторингу пошкодженого майна в Україні

Програма та методика тестування

КПІ.ІТ-9312.045440.04.51

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Ірина Вітковська

Нормоконтроль:

_____ Максим Головченко

Виконавець:

_____ Расім Ісмаїлов

Київ – 2023

ЗМІСТ

| | | |
|---|------------------------------------|---|
| 1 | ОБ'ЄКТ ВИПРОБУВАНЬ..... | 3 |
| 2 | МЕТА ТЕСТУВАННЯ | 4 |
| 3 | МЕТОДИ ТЕСТУВАННЯ..... | 5 |
| 4 | ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ | 6 |

| | | | | | | |
|------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.04.51 | Арк. |
| Змін | Арк. | № докум. | Підп. | Дата. | | 2 |

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Об'єктом випробування є програмне забезпечення для моніторингу пошкодженого майна в Україні.

| | | | | | | |
|------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.04.51 | Арк. |
| Змін | Арк. | № докум. | Підп. | Дата. | | 3 |

2 МЕТА ТЕСТУВАННЯ

Метою тестування є наступне:

- перевірка правильності роботи програмного забезпечення відповідно до функціональних вимог;
- перевірка збереження даних;
- пошук подій за ключовими словами;
- пошук подій за вказаними параметрами;
- подавання подій на розгляд;
- підтвердження чи відхилення поданих на розгляд подій менеджерами;
- реєстрація користувача;
- перевірка сумісності веб-додатку з останніми версіями сучасних браузерів (Chrome, Opera, Firefox, ...);
- знаходження проблем, помилок і недоліків з метою їх усунення;
- перевірка зручності графічного інтерфейсу.

| | | | | | | |
|------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.04.51 | Арк. |
| Змін | Арк. | № докум. | Підп. | Дата. | | 4 |

3 МЕТОДИ ТЕСТУВАННЯ

Для тестування програмного забезпечення використовуються такі методи:

– статичне тестування – перевіряється програма разом з усією документацією, яка аналізується на предмет дотримання стандартів програмування;

– динамічне тестування – застосовується в процесі виконання програми. Коректність програмного засобу перевіряється на певній кількості тестів. При прогоні кожного з них збираються та аналізуються дані про проблеми та помилки в роботі програми;

– функціональне тестування – полягає у перевірці відповідності реальної поведінки програмного забезпечення очікуваній;

– мануальне тестування – тестування без використання автоматизації, тест-кейси пише особа, що тестує програмне забезпечення.

| | | | | | | |
|------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.04.51 | Арк. |
| Змін | Арк. | № докум. | Підп. | Дата. | | 5 |

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується мануально з використанням наскрізного тестування, з метою знаходження помилок та недоліків як у функціональній частині програмного забезпечення так і в зручності користування. Для того, щоб перевірити працездатність та відмовостійкість застосунку, необхідно провести наступні тестування:

- динамічне тестування на відповідність функціональним вимогам;
- тестування на виведення повідомлень про помилку, коли це необхідно;
- тестування інтерфейсу користувача;
- тестування зручності використання.

| | | | | | | |
|------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.04.51 | Арк. |
| Змін | Арк. | № докум. | Підп. | Дата. | | 6 |

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ” _____ 2023 р.

Веб-застосунок для моніторингу пошкодженого майна в Україні

Керівництво користувача

КПІ.ІТ-9312.045440.05.34

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Ірина Вітковська

Нормоконтроль:

_____ Максим Головченко

Виконавець:

_____ Расім Ісмаїлов

Київ – 2023

ЗМІСТ

| | | |
|-----|--|----|
| 1 | ПРИЗНАЧЕННЯ ПРОГРАМИ | 3 |
| 2 | ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ..... | 4 |
| 2.1 | Системні вимоги для коректної роботи..... | 4 |
| 2.2 | Завантаження застосунку | 4 |
| 2.3 | Перевірка коректної роботи..... | 4 |
| 3 | ВИКОНАННЯ ПРОГРАМИ | 6 |
| 3.1 | Реєстрація | 6 |
| 3.2 | Авторизація | 7 |
| 3.3 | Пошук подій в обраній області | 7 |
| 3.4 | Пошук подій по обраним параметрам | 9 |
| 3.5 | Пошук подій за ключовими словами..... | 10 |
| 3.6 | Подавання інформації про обстріл на розгляд | 11 |
| 3.7 | Зміна персональних даних в особистому кабінеті | 12 |
| 3.8 | Вихід з аккаунту..... | 13 |
| 3.9 | Прийняття рішення щодо додавання нової події менеджером..... | 13 |

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.05.34 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 2 |

1 ПРИЗНАЧЕННЯ ПРОГРАМИ

Рішення для моніторингу пошкодженого майна в Україні являє собою веб-застосунок, який дозволяє оцінити ситуацію в різних областях, шукати події по заданим параметрам. За умови реєстрації, можна подавати на розгляд інформацію про випадки обстрілів власноруч.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.05.34 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 3 |

2 ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ

2.1 Системні вимоги для коректної роботи

Мінімальна конфігурація технічних засобів:

- тип процесору: Intel Core i5;
- об'єм ОЗП: 4 Гб;
- підключення до мережі Інтернет зі швидкістю від 20 мегабіт;

Рекомендована конфігурація технічних засобів:

- тип процесору: Intel Core i5;
- об'єм ОЗП: 8 Гб;
- підключення до мережі Інтернет зі швидкістю від 100 мегабіт;

2.2 Завантаження застосунку

На даний момент застосунок можна розгорнути локально, слідуючи інструкціям з відповідного розділу пояснювальної записки. Завантажити застосунок можна за адресою <https://github.com/Rasim-I/Destruction-Map>.

2.3 Перевірка коректної роботи

Після розгортання застосунку і переходу по одній з запропонованих адрес в браузері, користувач має побачити головну сторінку, яка показана на рисунку 2.1.

| | | | | | | |
|-------|------|----------|-------|-------|--------------------------|------|
| | | | | | КПІ.ІТ-9312.045440.05.34 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 4 |


Select building type:

- Residential
- Industrial
- Administrative
- Healthcare
- School or childcare
- Military
- Commercial
- Religious
- Cultural
- Transport
- Humanitarian
- Food Infrastructure

Select weapon system:

- HE rocket artillery
- Cluster munition
- Incendiary munition
- Cruise missile
- Ballistic missile
- Thermobaric munition
- Vehicle mounted weapon
- Small arms
- Land mines
- Anti air missile
- Loitering munition

Date:

dd-----yyyy 

Location

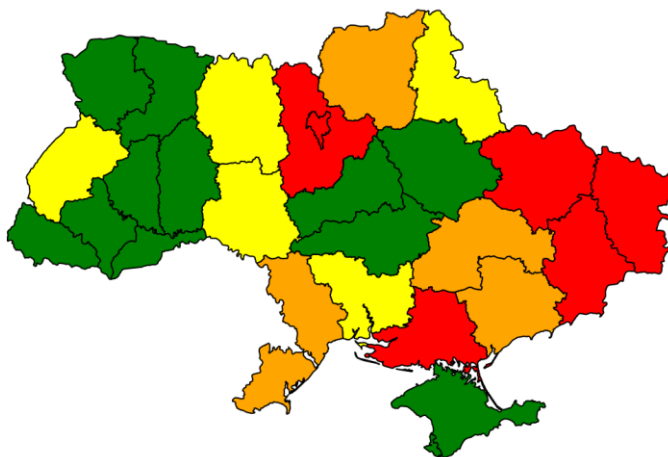


Рисунок 2.1 – Головна сторінка застосунку

| | | | | |
|-------|------|----------|-------|-------|
| | | | | |
| Змін. | Арк. | № докум. | Підп. | Дата. |

КПІ.ІТ-9312.045440.05.34

3 ВИКОНАННЯ ПРОГРАМИ

3.1 Реєстрація

Щоб користувач отримав змогу використовувати весь функціонал веб-застосунку, йому потрібно зареєструватись. Це можна зробити, натиснувши на кнопку особистого кабінету в правому верхньому куті сторінки, як показано на рисунку 3.1.



Рисунок 3.1 – Кнопка особистого кабінету користувача

Користувач опиниться на сторінці авторизації, яка показана на рисунку 3.2 і натиснути кнопку реєстрації.

Рисунок 3.2 – Вигляд форми авторизації

Сторінка реєстрації показана на рисунку 3.3.

Register

| |
|------------------|
| Name |
| Surname |
| Address |
| Age |
| Email |
| Password |
| Confirm Password |

Register

© 2023 - Destruction Map - [Privacy](#)

Рисунок 3.3 – Вигляд сторінки реєстрації

Користувачу потрібно заповнити поля для вводу особистих даних і натиснути кнопку підтвердження реєстрації. Якщо всі поля заповнено і введені дані відповідають очікуваному формату – реєстрація пройде успішно і користувач потрапить на головну сторінку.

3.2 Авторизація

Щоб авторизуватись зареєстрованому користувачу потрібно перейти з головної сторінки на сторінку авторизації, яка показана на рисунку 3.2. Потрапити на неї можна, натиснувши кнопку особистого кабінету, рисунок 3.1.

3.3 Пошук подій в обраній області

Користувач натискає на одну з областей на карті, представлений на головній сторінці, рисунок 3.4

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.05.34 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 7 |

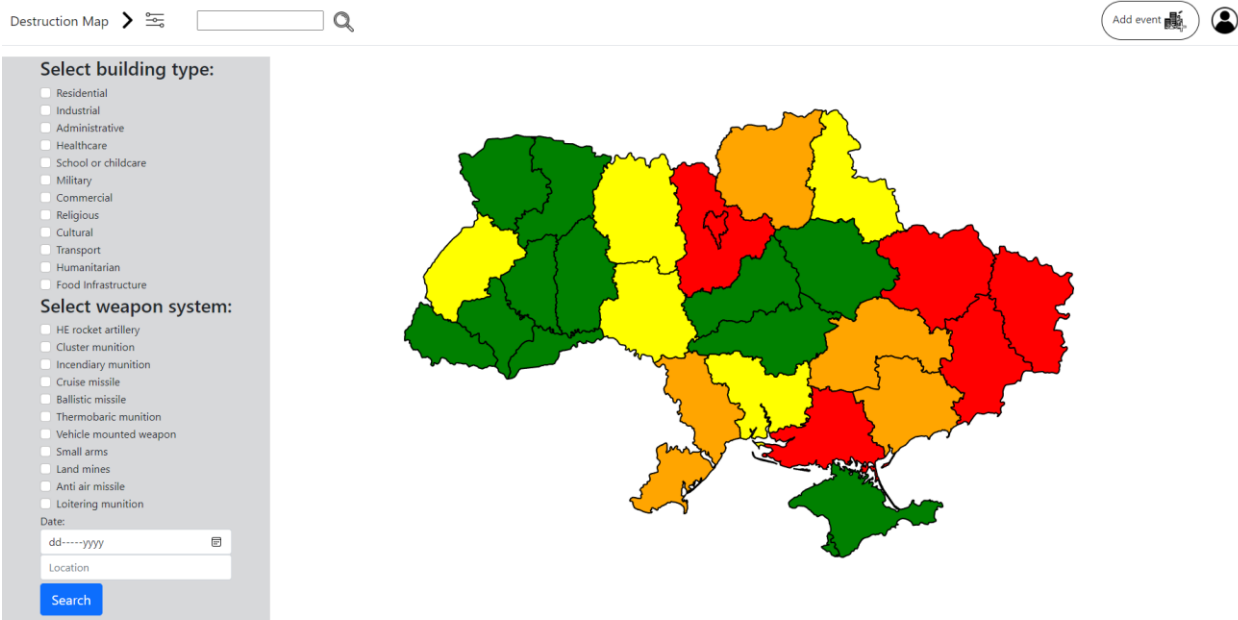


Рисунок 3.4 – Области карти на головній сторінці

Якщо інформація по подіям в певній області присутня – користувач побачить список подій в лівій частині екрану, як на рисунку 3.5.

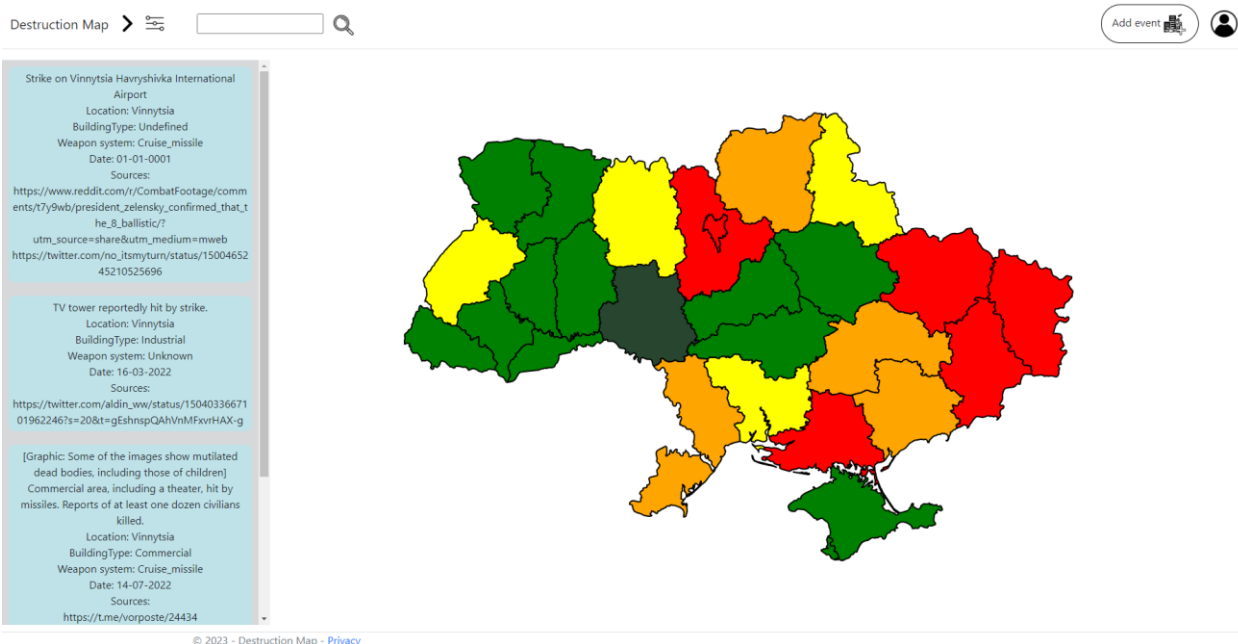


Рисунок 3.5 – Результат пошуку подій по обраній області

| | | | | | | | | | | | |
|-------|------|----------|-------|-------|--|--|--|--|--|--------------------------|------|
| | | | | | | | | | | КПІ.ІТ-9312.045440.05.34 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | | | | | | 8 |

3.4 Пошук подій по обраним параметрам

Щоб відкрити меню вибору параметрів, користувачу потрібно натиснути на кнопку в верхній лівій частині сторінки, яка виглядає як три горизонтальні лінії. Показано на рисунку 3.6.

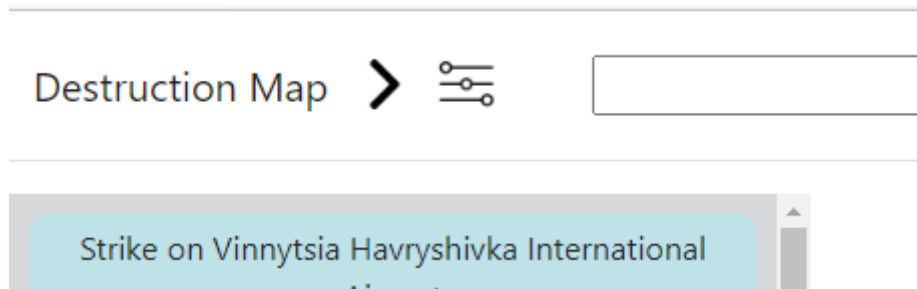


Рисунок 3.6 – Частина сторінки, на якій зображена кнопка відкриття меню параметрів

Далі користувач побачить меню в лівій частині сторінки, як показано на рисунку 3.7.

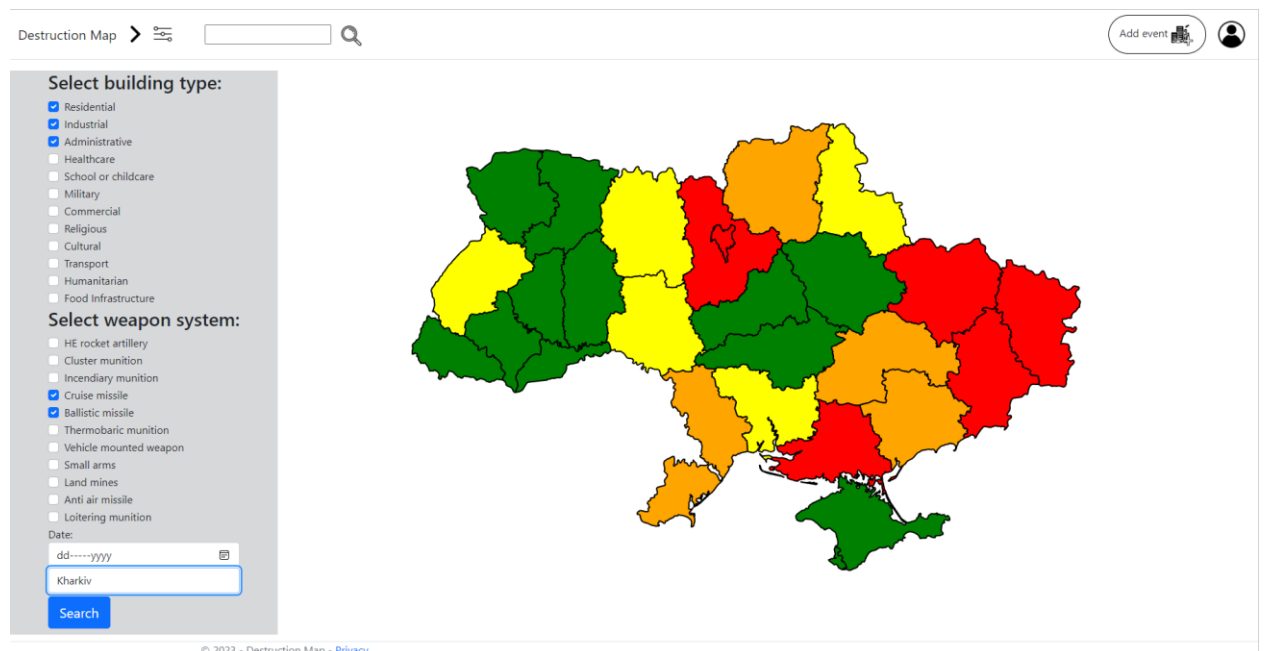


Рисунок 3.7 – Головна сторінка з відкритим меню параметрів

Користувач обирає бажані параметри, такі як тип пошкодженої будівлі, тип використаної зброї, дату події та місце. Після натискання кнопки

пошуку, меню вибору параметрів зміниться на список подій, знайдених по заданим параметрам, як на рисунку 3.8.

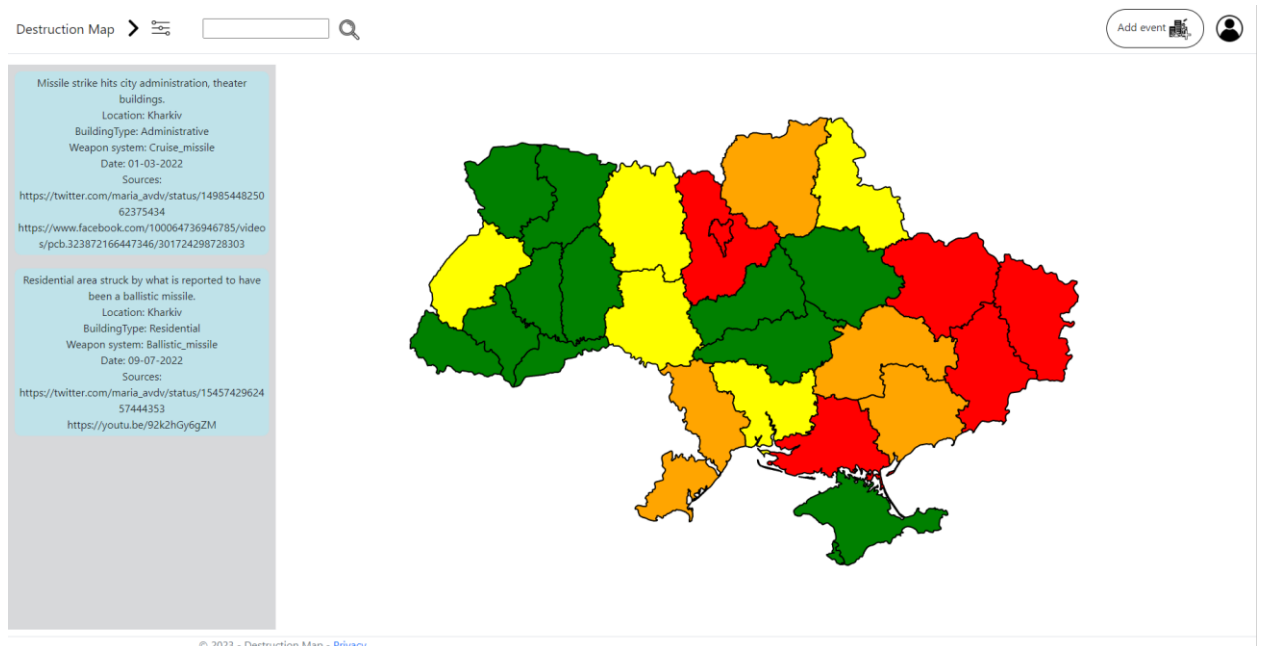


Рисунок 3.8 – Результат пошуку по заданим параметрам

3.5 Пошук подій за ключовими словами

На головній сторінці користувачу треба ввести ключові слова в поле для пошуку по ключовим словам, яке показано на рисунку 3.9.

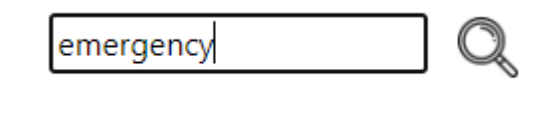


Рисунок 3.9 – Поле для пошуку по ключовим словам з введеним пошуковим запитом

Після натискання на кнопку пошуку, яка знаходиться зліва від поля вводу пошукового запиту, користувач побачить результат в лівій частині сторінки, як на рисунку 3.10.

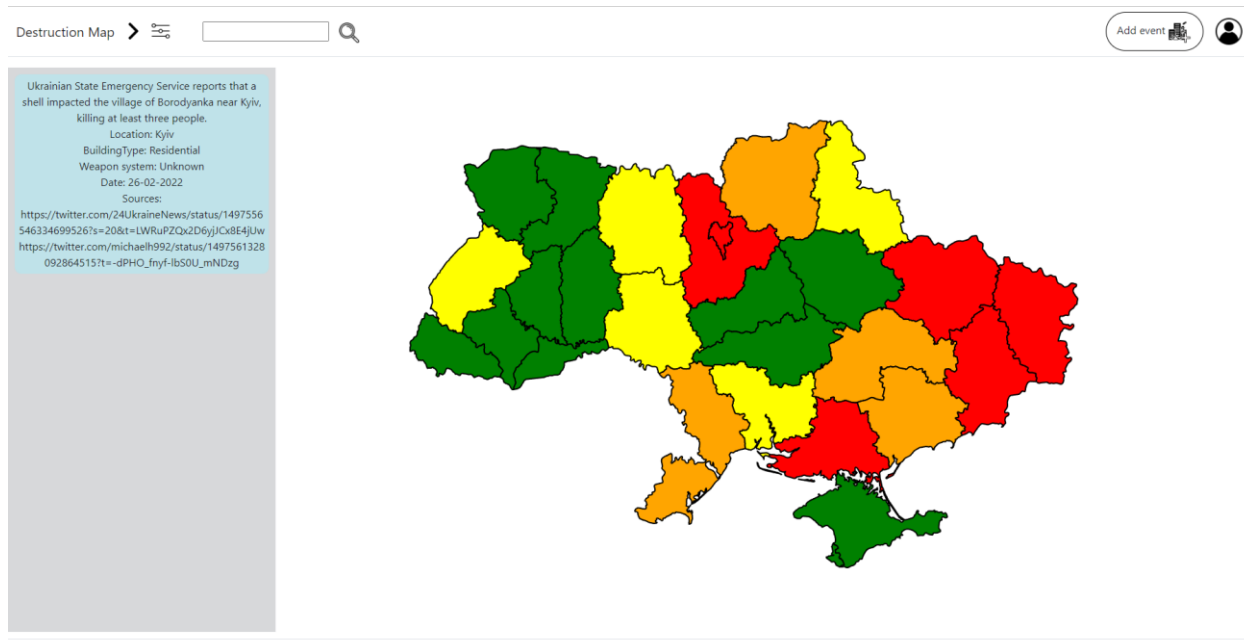


Рисунок 3.10 – Результат пошуку по ключовим словам

3.6 Подавання інформації про обстріл на розгляд

Авторизований користувач може подати інформацію про обстріл на розгляд менеджером застосунку. Щоб це зробити, користувач має натиснути на кнопку для сповіщення про подію, яка показана на рисунку 3.11.



Рисунок 3.11 – Кнопка переходу на сторінку сповіщення про подію

На сторінці додавання інформації про подію (рисунок 3.12), користувачу треба заповнити поля та натиснути кнопку відправки. Якщо не всі поля будуть заповнені, користувачу буде про це сповіщено і він не зможе відправити форму.

Add event

Event date:

© 2023 - Destruction Map - [Privacy](#)

Рисунок 3.12 – Сторінка додавання інформації про подію

3.7 Зміна персональних даних в особистому кабінеті

Щоб змінити персональні дані, авторизованому користувачу потрібно перейти в особистий кабінет, натиснувши кнопку, яка показана на рисунку 3.1. Він потрапить на сторінку особистого кабінету, де зможе ввести бажані зміни і зберегти зміни. Сторінка особистого кабінету показана на рисунку 3.13.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.05.34 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 12 |

User Page

Email:
user1@mai.com

Name: Petro
Name

Surname: User
Surname

Address: Odessa
Address

Save

Рисунок 3.13 – Сторінка особистого кабінету

3.8 Вихід з аккаунту

Щоб вийти з аккаунту авторизованому користувачу треба натиснути кнопку «Logout» в верхньому правому куті сторінки. Кнопка виходу з аккаунту показана на рисунку 3.14.



Рисунок 3.14 – Кнопка виходу з аккаунту

3.9 Прийняття рішення щодо додавання нової події менеджером

Якщо користувач має роль менеджера, він може перейти на сторінку огляду інформації про нові події, додані авторизованими користувачами. Щоб потрапити на цю сторінку, менеджер має натиснути кнопку «Approve Event», як на рисунку 3.15.

| | | | | | | |
|-------|------|----------|-------|-------|-------------------------|------|
| | | | | | КП.ІТ-9312.045440.05.34 | Арк. |
| Змін. | Арк. | № докум. | Підп. | Дата. | | 13 |



Рисунок 3.15 – Інтерфейс менеджера

Сторінка для огляду подій і прийняття рішень по ним виглядає, як показано на рисунку 3.16.

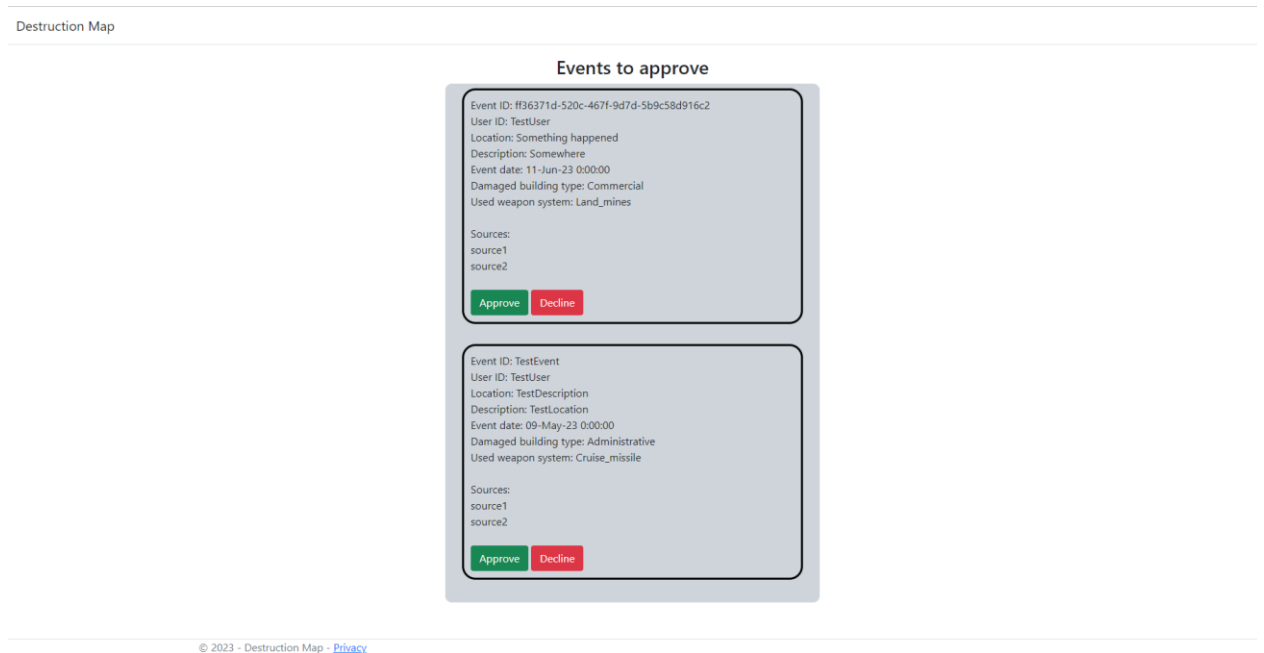


Рисунок 3.16 – Сторінка прийняття рішень по поданим на розгляд подіям

Якщо менеджер вважає, що подія має бути додана до основного списку, він натискає кнопку «Approve». Якщо він не вважає, що ця подія має бути додана, він натискає кнопку «Decline».

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ” _____ 2023 р.

Веб-застосунок для моніторингу пошкодженого майна в Україні

Графічний матеріал

КПІ.ІТ-9312.045440.06.99

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Ірина Вітковська

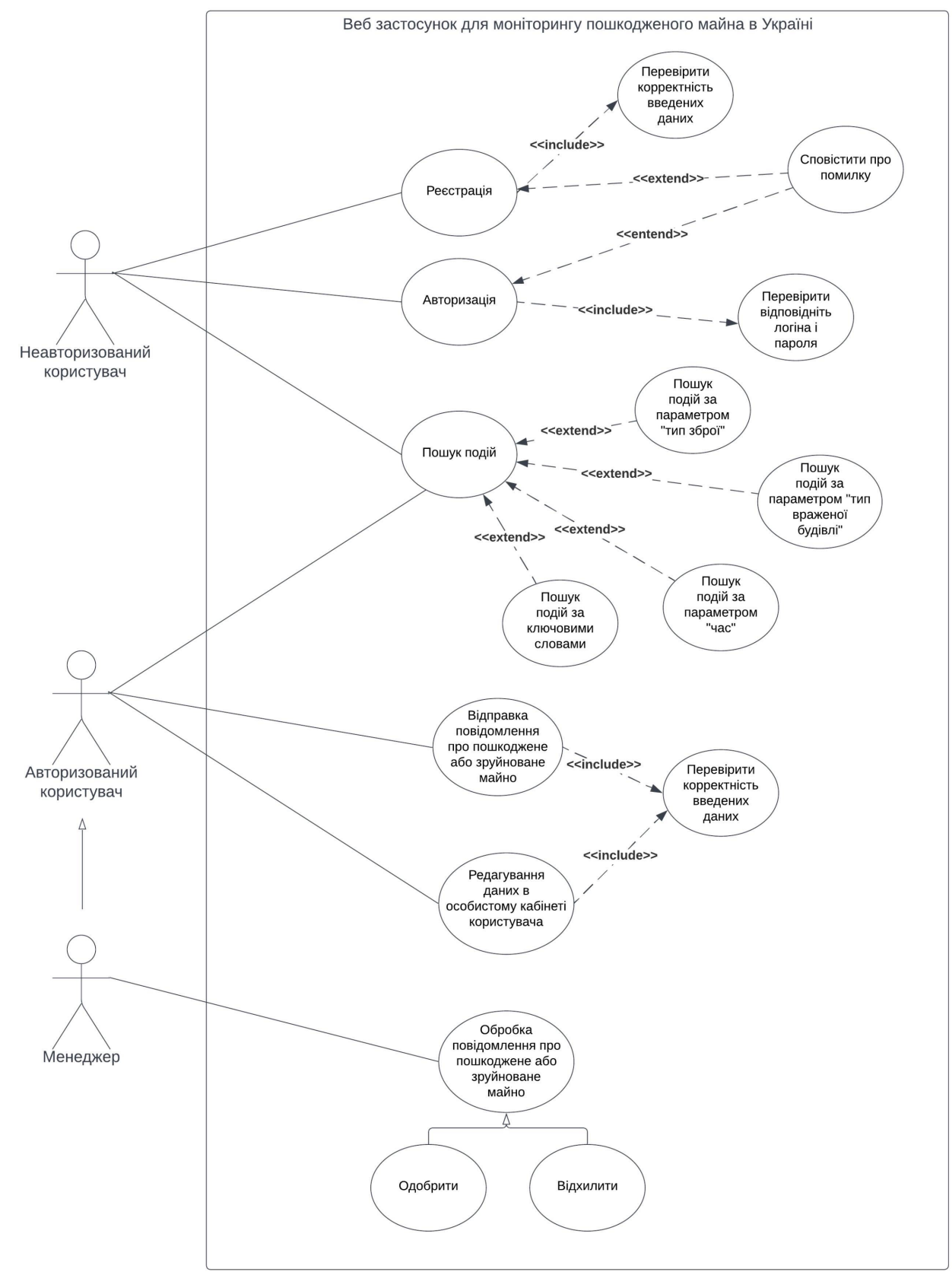
Нормоконтроль:

_____ Максим Головченко

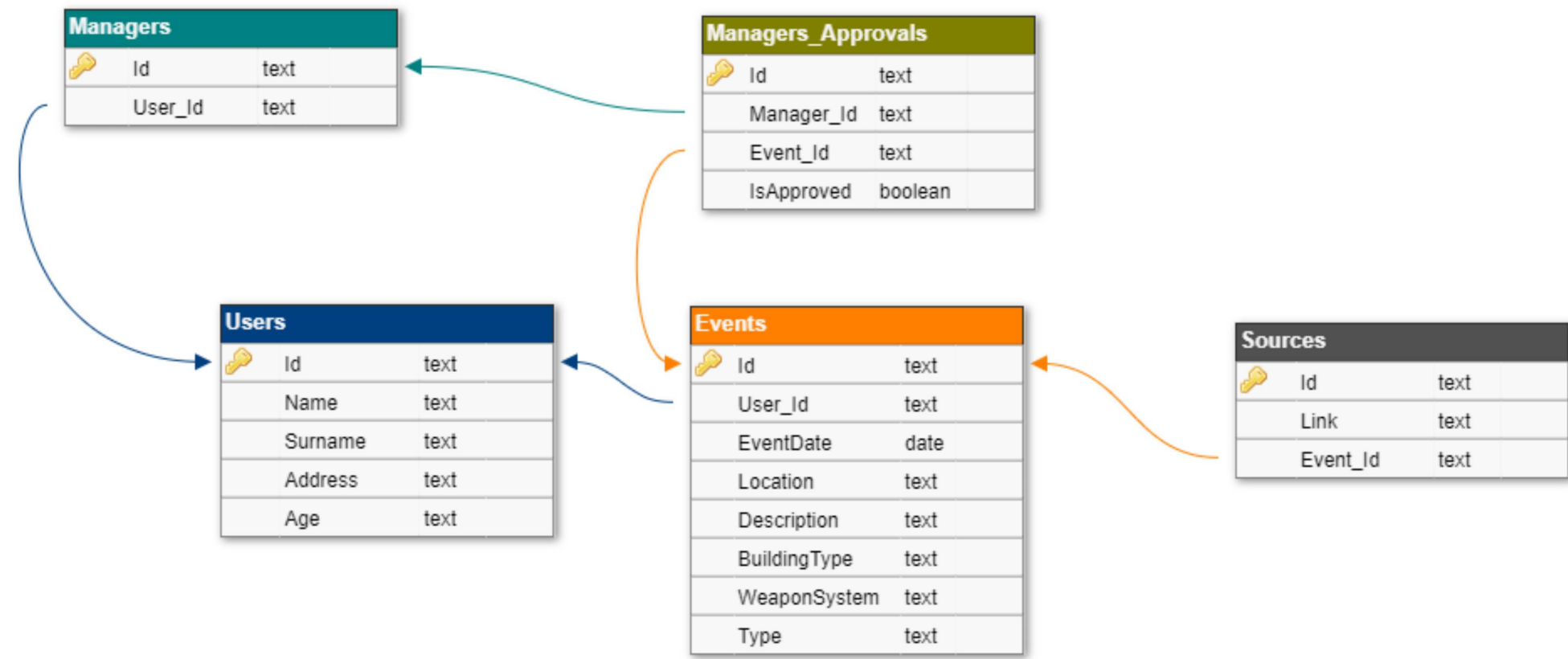
Виконавець:

_____ Расім Ісмаїлов

Київ – 2023



| | | | | | | | |
|-----------|------|------------------|--------|------|---|---------|--|
| | | | | | КПІ.ІТ-9312.045440.06.99.ССВВ | | |
| | | | | | Структурна схема варіантів використання | | |
| Зм. | Арк. | № документа | Підпис | Дата | | | |
| | | | | | | | |
| Розробив | | Ісмаїлов Р. Т. | | | | | |
| Перевірів | | Вітковська І. І. | | | | | |
| Т. кон. | | | | | Аркуш | Аркушів | |
| Н. кон. | | Головченко М. М. | | | КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-93 | | |
| Затвердив | | Вітковська І. І. | | | | | |
| | | | | | Веб-застосунок для моніторингу пошкодженого майна в Україні | | |



| | | | | | | | | | | |
|-----------|------|------------------|--------|------|---|--|--|--|---------|---------|
| | | | | | КПІ.ІТ-9312.045440.07.99.СБД | | | | | |
| | | | | | Схема бази даних | | | Літера | Маса | Масштаб |
| Зм. | Арк. | № документа | Підпис | Дата | | | | | | |
| Розробив | | Ісмаїлов Р. Т. | | | | | | | | |
| Перевірів | | Вітковська І. І. | | | | | | | | |
| Т. кон. | | | | | | | | Аркуш | Аркушів | |
| Н. кон. | | | | | | | | КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-93 | | |
| Затвердив | | | | | | | | | | |
| | | | | | Веб-застосунок для моніторингу пошкодженого майна в Україні | | | | | |

User Page

Email:

Name: Anton

Surname: User

Address: Dnipro

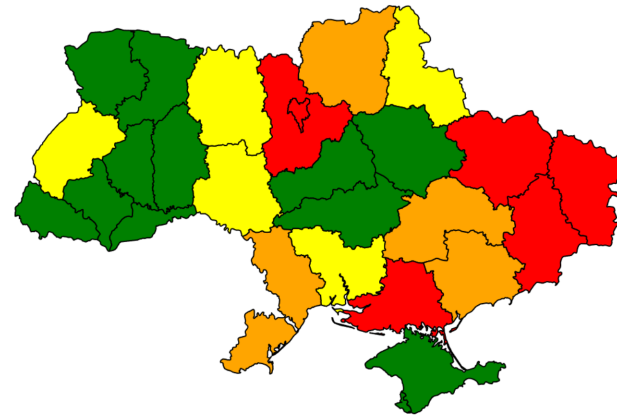
Select building type:

- Residential
- Industrial
- Administrative
- Healthcare
- School or childcare
- Military
- Commercial
- Religious
- Cultural
- Transport
- Humanitarian
- Food Infrastructure

Select weapon system:

- HE rocket artillery
- Cluster munition
- Incendiary munition
- Cruise missile
- Ballistic missile
- Thermobaric munition
- Vehicle mounted weapon
- Small arms
- Land mines
- Anti air missile
- Loitering munition

Date:



Events to approve

Event ID: ff36371d-520c-467f-9d7d-5b9c58d916c2

User ID: TestUser

Location: Something happened

Description: Somewhere

Event date: 11-Jun-23 00:00:00

Damaged building type: Commercial

Used weapon system: Land_mines

Sources:
source1
source2

Event ID: TestEvent

User ID: TestUser

Location: TestDescription

Description: TestLocation

Event date: 09-May-23 00:00:00

Damaged building type: Administrative

Used weapon system: Cruise_missile

Sources:
source1
source2

Log in

Remember me?

[Register as a new user](#)

Rocket motor embedded in pavement in residential area.

Location: Bucha, Kyiv vicinity

BuildingType: Residential

Weapon system: Cruise_missile

Date: 25-02-2022

Sources:
<https://twitter.com/UAWeapons/status/1497176385420464133>
<https://twitter.com/3TAmsL026aRar/status/1497161918586499109>

Missile strike hits city administration, theater buildings.

Location: Kharkiv

BuildingType: Administrative

Weapon system: Cruise_missile

Date: 01-03-2022

Sources:
https://twitter.com/maria_jardj/status/1498544825062375434
https://www.facebook.com/100064736946785/vid_eos/pcb.323872166447346/301724298728303

Remnants of a missile at an oil depot

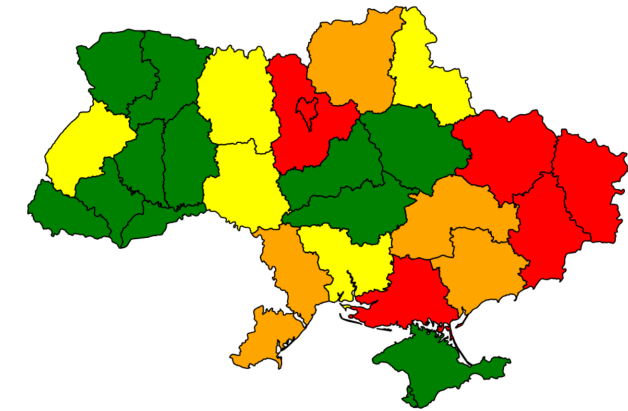
Location: Kirov, Donetsk

BuildingType: Industrial

Weapon system: Ballistic_missile

Date: 25-02-2022

Sources:
https://twitter.com/DNR_online/status/1497495576618934272?s=20&t=gn_00P9gSmUzJkmsNup8Cg



Add event

Event date:

-- Select building type --

-- Select weapon system --

| | | | | | | | | |
|-----------|------|------------------|--------|------|---|--------|---------|---------|
| | | | | | <i>KPI.IT-9312.045440.10.99.KE</i> | | | |
| | | | | | Креслення виду екранних форм | Літера | Маса | Масштаб |
| Зм. | Арк. | № документа | Підпис | Дата | | | | |
| Розробив | | Ісмаїлов Р. Т. | | | | | | |
| Перевірив | | Вітковська І. І. | | | Веб-застосунок для моніторингу пошкодженого майна в Україні | Аркуш | Аркушів | |
| Т. кон. | | | | | | | | |
| Н. кон. | | Головченко М. М. | | | КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-93 | | | |
| Затвердив | | Вітковська І. І. | | | | | | |