

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Олександр Коваль

«__» _____ 2020 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні технології моніторингу
довкілля»

спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «Система обліку надзвичайних подій соціально-політичного
характеру»

Виконала:

студентка IV курсу, групи ТМ-62

Конопльова Софія Вікторівна _____

Керівник:

Ст. викладач Шульженко Олег Феодосійович _____

Рецензент:

Головний інженер науково-інженерного центру

“Водоекологія” к.т.н. Писарук Віктор Іванович _____

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

Студентка _____

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 122 Комп'ютерні науки та інформаційні технології

Спеціалізація Інформаційні технології моніторингу довкілля

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр Коваль
(підпис)

” ____ ” _____ 2020р.

ЗАВДАННЯ

на дипломну роботу студенту

Конопльовій Софії Вікторівні

(прізвище, ім'я, по батькові)

1. Тема роботи Система обліку надзвичайних подій соціально-політичного характеру

керівник роботи Ст. викладач Шульженко Олег Феодосійович
(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ”25” травня 2020р. № 1168-с

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи База даних, а також ASP.Net Core проект.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати існуючі програмні рішення та засоби обліку надзвичайних подій соціально-політичного характеру, спроектувати архітектуру системи, розробити веб-систему з інтерактивною картою, розробити інтерфейс адміністратора.

5. Перелік ілюстративного матеріалу

1. Мета роботи 2. Актуальність 3. Діаграма прецедентів системи 4. Модель бази даних 5. Перегляд інформації на інтерактивній карті 6. Перегляд інформації по таблицях 7. Розрахунок збитків 8. Вхід до кабінету адміністратора 9. Додання нової надзвичайної ситуації 10. Редагування даних надзвичайної ситуації 11. Висновки

6. Дата видачі завдання "11" _____ жовтня _____ 2019 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	25.10.19	
2.	Вивчення та аналіз задачі	03.02-11.02.20	
3.	Розробка архітектури та загальної структури системи	12.02-20.02.20	
4.	Розробка структур окремих підсистем	21.02-28.02.20	
5.	Програмна реалізація системи	02.03-11.03.20	
6.	Оформлення пояснювальної записки	12.03-29.05.20	
7.	Захист програмного продукту	05.06.20	
8.	Передзахист	10.06.20	
9.	Захист	16.06.20	

Студент _____

(підпис)

Конопльова С. В. _____

(прізвище та ініціали,)

Керівник роботи _____

(підпис)

Шульженко О. Ф. _____

(прізвище та ініціали,)

АНОТАЦІЯ

Метою роботи є створення веб-системи обліку надзвичайних ситуацій соціально-політичного характеру з інтерактивною картою України. Система надає звичайним користувачам можливість дослідити дані виниклих надзвичайних ситуацій по різних регіонах України у різні роки. Адміністратор має доступ до створення нових маркерів на карті з внесенням інформації про нові надзвичайні ситуації, а також редагування значення існуючих маркерів.

Записка містить 61 сторінки, 26 рисунків та 38 посилань.

ABSTRACT

The purpose of this work is to create a web-based system for managing socio-political emergencies data with an interactive map of Ukraine. The system provides regular users with the opportunity to study the data of emergency situations that occurred in different regions of Ukraine in different years. Administrator has access to creating new markers on the map with information about new emergencies, as well as changing the value of existing markers.

The note contains 61 pages, 26 illustrations and 38 references.

ЗМІСТ

Вступ	8
1. Постановка задачі розробки інтерактивної карти надзвичайних ситуацій соціально-політичного характеру в Україні	10
2. Аналіз проблеми розробки інтерактивної карти надзвичайних подій	12
2.1. Сутність обліку надзвичайних ситуацій соціально-політичного характеру	12
2.1.1. Епідемії	13
2.1.2. Вибухи, розливи токсичних речовин та пожежі	14
2.1.3. Боротьба з надзвичайними ситуаціями	15
2.2. Збір інформації	18
2.2.1. Інформаційна потреба	19
2.2.2. Об'єднання інформації з декількох джерел	20
2.2.3. Захист від дезінформації та інших видів шкідливого контенту ...	21
2.3. Вимоги до розробки	21
2.3.1. Метод доступу	22
2.3.2. Модель даних	22
2.4. Інтерактивні карти в Україні та світі	23
2.4.1. Українська платформа Liveuamap	24
2.4.2. Угорська система RSOE — Emergency and Disaster Information Service	25
2.4.3. Copernicus Emergency Management Service	26
2.4.4. PDC DisasterAWARE	28
3. Засоби розробки	30
3.1. Середовище розробки Visual Studio / VS Code	30
3.2. Мова програмування C#	30
3.3. Середовище CLR	31

3.4. Платформа .NET Core	31
3.5. Служба Azure Web Apps	32
3.6. SQL Server (SQL Azure)	32
3.7. JWT автентифікація	33
3.8. JavaScript	33
3.9. Фреймворк Angular	33
3.10. Google Maps API	34
4. Опис програмної реалізації	36
4.1. Опис функціональності системи	36
4.2. Модель бази даних	38
4.3. Проектування архітектури системи	41
4.4. Реалізація серверної частини системи	45
5. Робота користувача з програмною системою	47
Висновки	57
Список використаних джерел	58
Додаток А	62
Додаток Б	64
Додаток В	75

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

НС — надзвичайна ситуація;

ГІС — геоінформаційна система;

БД — база даних;

API (англ. Application Programming Interface) — прикладний програмний інтерфейс;

ВСТУП

У наш час проблеми безпеки людини і суспільства в цілому стають особливо актуальними. Кожен день в Україні та світі реєструються десятки тисяч надзвичайних ситуацій, які порушують нормальні умови життя та діяльності людей і призводять до загроз життю або здоров'ю населення, великої кількості смертей і травм, що призводить до значних матеріальних втрат.

Останніми десятиліттями зростає кількість НС соціально-політичного характеру. Такі НС пов'язані із погрозами чи здійсненням терористичних актів та діями антиконституційного спрямування, викраденням або спробою викрадення зброї, захопленням заручників тощо [1].

Прикладів подібних дій за останні роки в Україні предостатньо. Це і терористичні акти, підриви військових складів з боєприпасами, і захоплення складів зі зброєю і постійні мінування транспортних засобів, громадських установ та систем життєзабезпечення.

На сьогодні практично немає суто українських інформаційно-аналітичних систем на базі картографічних веб-платформ, призначених для обліку надзвичайних ситуацій.

Тому потрібно розробити веб-систему, яка використовуючи дані про НС, дозволить підвищити обізнаність людей про можливі наслідки тих, чи інших дій. Система повинна звернути увагу громадськості на НС, особливо коли вони можуть стати причиною великої кількості постраждалих і привести до загрози нормальному життю всього регіону або країни. Також система може бути взятою до уваги уповноваженими органами державної влади, а також бути використаною дослідниками та незалежними журналістами, які шукають точну інформацію.

Користувачем даної інформаційно-аналітичної системи може бути як кожен відвідувач сайту, який може переглянути довідку про виникнення надзвичайних ситуацій в конкретних регіонах України за певні роки, так і адміністратор, який буде

вести облік НС, проводити облік вжитих заходів для подолання та ліквідації наслідків НС, розраховувати збитки від НС та заносити в базу даних і відображати на карті всі точки НС, їх координати.

Кожен користувач системи отримує можливість на карті України вибирати будь-яку точку та завдяки даним цієї області переглянути результати пошуку за певний проміжок часу в конкретному регіоні.

Звіт містить 5 розділів.

У першому розділі описується постановка задачі розробки інтерактивної карти надзвичайних ситуацій соціально-політичного характеру в Україні.

У другому розділі описується аналіз проблеми розробки інтерактивної карти.

У третьому розділі описуються основні засоби розробки даної системи.

У четвертому розділі вказується опис програмної реалізації системи.

У п'ятому розділі описується робота користувача з програмною системою.

1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ІНТЕРАКТИВНОЇ КАРТИ НАДЗВИЧАЙНИХ СИТУАЦІЙ СОЦІАЛЬНО- ПОЛІТИЧНОГО ХАРАКТЕРУ В УКРАЇНІ

В сучасному світі люди звикли засвоювати інформацію не у вигляді набору деяких даних на папері, а за допомогою візуалізації даних на екранах електронних пристроїв. З темпами розвитку технологій зростає роль інформаційно-аналітичних систем, адже дуже важливим є систематизування великих масивів даних.

Особливість використання картографії в дослідженні географічних аспектів соціально-політичної безпеки полягає в тому, що вона дозволяє географії за допомогою карт, дати узагальнення сучасного рівня інформації та знань про соціально-політичну безпеку України, дослідити особливості надзвичайних ситуацій в Україні та їхні зміни в часі [2].

Візуалізація системи надзвичайних ситуацій соціально-політичного характеру в Україні, орієнтована на відображення даних про надзвичайні ситуації, що виникли, у вигляді інтерактивної карти України, надає зручне отримання та швидше розуміння інформації у межах певного регіону.

Задача візуалізації системи вимагає створення інформаційної бази даних та її наповнення інформацією про виниклі надзвичайні ситуації соціально-політичного характеру, прив'язку подій до координат місцевості, де вони виникли та розробку веб-системи із зрозумілим інтерфейсом користувача.

Звичайний користувач системи не повинен вводити ніякі додаткові вхідні дані. Йому потрібно лише вибрати на карті точку, натиснувши на неї мишкою, щоб отримати інформацію про надзвичайну ситуацію у тій місцевості. Також у системі можна переглянути розширені дані у формі таблиць. Завдяки простому та інтуїтивному інтерфейсу системи користувачу не потрібні мати спеціальні знання.

Окрім звичайних користувачів, у системі передбачена авторизація адміністратора веб-системи. Він має доступ до додавання нових надзвичайних

ситуацій. Також передбачено доступ до редагування існуючих даних у разі виникнення помилки.

Розроблена веб-система повинна давати можливість:

- вести облік НС;
- заносити в базу даних та відображати на карті всі точки НС, їх координати;
- переглянути результати пошуку за певний проміжок часу в конкретному регіоні;
- проводити облік вжитих заходів для подолання та ліквідації наслідків НС;
- розраховувати збитки від НС.

2. АНАЛІЗ ПРОБЛЕМИ РОЗРОБКИ ІНТЕРАКТИВНОЇ КАРТИ НАДЗВИЧАЙНИХ ПОДІЙ

Ефективний облік стихійних лих і реагування на них вимагають швидкого використання інформації і даних з багатьох джерел. Здатність легко інтегрувати і розподіляти цифрові дані в просторово явні форми для швидкої оцінки та аналізу під час і після стихійного лиха залишається складним завданням. Спеціалізовані дані, мережі передачі даних, методи і технології обробки інформації необхідні в умовах надзвичайної ситуації. Одним із пріоритетів у цій ситуації є створення ефективної системи оповіщення населення про надзвичайну ситуацію.

Після надзвичайної ситуації існує значний попит і потреба в картах і просторової інформації. Для здійснення ефективного реагування співробітники служб реагування на надзвичайні ситуації повинні отримувати інформацію про місцезнаходження постраждалого населення і про те, як найшвидше дістатися до нього [3].

Алан Макікрен та ін. виявили, що "соціальні мережі стають все більш географічними", і потенційне використання цієї інформації в надзвичайних ситуаціях було визнано як академічними інститутами, так і організаціями з надзвичайних ситуацій [4].

2.1. Сутність обліку надзвичайних ситуацій соціально-політичного характеру

В середньому за один день в світі "відбувається від двох до трьох катастроф в їх надзвичайному стані, від 15 до 20 в їх відновних періодах і близько десятка активних надзвичайних ситуацій, пов'язаних з конфліктами" [5]. До них відносяться такі події, як терористичні акти, підриви військових складів з боєприпасами, постійні мінування транспортних засобів, громадських установ та систем життєзабезпечення.

У поєднанні з втратою зв'язку це стає періодом невизначеності, який може викликати колективний стрес, що виникає в результаті позбавлення значної частини суспільства очікуваних звичайних умов життя. Ці умови соціально визначаються як нормальні людські потреби [6].

Однак широка громадськість часто бере активну участь у катастрофах, і їх участь стає все більш помітною завдяки використанню інформаційно-комунікаційних технологій [7].

На відміну від багатьох НС, наприклад, природного характеру, більшість соціально-політичних небезпек можна було б запобігти, скоротивши або навіть усунувши людські жертви і шкоду майну. При більш глибокому розумінні обставин, що викликають НС, ми можемо працювати над пом'якшенням і, можливо, ліквідацією деяких з них.

2.1.1. Епідемії.

Епідеміологи використовують карти для реєстрації місця розташування і вивчення поширення хвороби [8]. Додайте до карти здатність проводити просторовий аналіз за допомогою досягнень в області географічних інформаційних інструментів, і в результаті ви отримаєте технологію, яка добре підходить для відстеження захворювань.

Дослідження, які кількісно визначають небезпеку свинцю, моделюють вплив електромагнітних полів і відстежують хвороби, що переносяться повітрям і водою, — все це приносить користь від розвитку технологій в області географічної інформатики.

Геоінформаційна система була використана для виявлення та локалізації факторів екологічного ризику, пов'язаних з хворобою Лайма, в окрузі Балтімор, штат Меріленд [9]. Дані про водозбір, землекористування, тип ґрунту, геологію та розподіл лісів були зібрані в резиденціях пацієнтів з хворобою Лайма і об'єднані з даними, зібраними за випадково вибраними адресами, щоб створити модель, що визначає найбільш ймовірні місця, де може виникнути хвороба Лайма.

За допомогою геоінформаційних систем набагато простіше об'єднати епідеміологічні дані та екологічні дані для моделювання та прогнозування поширення та передачі хвороб. Ця інтеграція даних має важливе значення, якщо ми сподіваємося пом'якшити епідемії за рахунок більш ефективного планування політики в галузі охорони здоров'я.

На національному рівні геоінформаційна система була використана для розробки системи спостереження для моніторингу та боротьби з малярією в Ізраїлі [10]. Система спостереження на основі геоінформаційної системи визначила місця розмноження комарів *Anopheles*, завезені випадки малярії та населені пункти, щоб краще реагувати на випадки спалахів захворювання.

У глобальному масштабі, Національне управління з аеронавтики і дослідження космічного простору (НАСА) створило програму глобального моніторингу та прогнозування захворювань в Дослідницькому центрі Еймса для виявлення факторів навколишнього середовища, що впливають на характер ризику захворювання і його передачі. Програма розробила прогностичні моделі динаміки популяції переносників і ризику передачі хвороб з використанням даних дистанційного зондування і ГІС-технологій і застосувала їх для епіднагляду і боротьби з малярією [11].

2.1.2. Вибухи, розливи токсичних речовин та пожежі.

НС соціально-політичного характеру — це екстремальні події, які можуть бути випадковими, наприклад розливи токсичних речовин, або навмисними, наприклад вибухи бомб, приведених в дію терористами. Незалежно від їх походження, багато з цих НС можуть бути зменшені або навіть запобігти їм шляхом інтеграції геоінформаційних технологій. Наприклад, у разі викиду токсичних речовин дані про населення, місцезнаходження житлових об'єктів, швидкості і напрямку вітру можуть бути використані для побудови моделі, що дозволяє скласти карту масштабів лиха і запропонувати стратегії евакуації.

Всі кризи вимагають негайного і добре скоординованого реагування там, де обробка даних і взаємодія систем мають вирішальне значення. Крім політичних і

технічних проблем, пов'язаних зі злиттям даних зі змішаних джерел, власні формати даних часто перешкоджають взаємодії.

Загальні елементи, такі як системи координат і представлення місць розташування, кордонів і повітряних об'єктів, повинні виходити за рамки широкого співтовариства користувачів і постачальників інструментів.

Метадані мають вирішальне значення для ідентифікації та управління якістю даних, в той час як протоколи доступу і параметри пошуку визначатимуть швидкість вилучення даних з цифрового архіву або бібліотеки.

Під час кризи, такої як вибух в Оклахома-Сіті, швидкість доступу до даних має першорядне значення для рятувальників, і досягнення в області надійних механізмів індексації, таких як ГІС, виявляться неоціненними.

Хоча багато успішних ініціатив вже здійснюються як на місцевому, так і на національному рівнях, вони могли б значною мірою скористатися досягненнями в галузі географічної інформатики.

У місті Вінстон-Салем, штат Північна Кароліна, була побудована інтегрована мережева система пожежних операцій (I.N.F.O.), яка призначена для скорочення часу, необхідного пожежним для реагування на екстрені виклики та надання інформації про адресу інциденту, щоб допомогти пожежникам у прийнятті більш обґрунтованих рішень і плануванні зусиль з гасіння пожежі на шляху прямування [12]. I.N.F.O. автоматично використовує адресу інциденту для пошуку будь-якої попередньої інформації планування, яка може бути доступна (наприклад, плани поверхів будівель, мешканці, інформація про небезпечні відходи).

2.1.3. Боротьба з надзвичайними ситуаціями.

Боротьба з надзвичайними ситуаціями — це процес планування і вжиття заходів щодо мінімізації соціальних і фізичних наслідків НС та зниження вразливості населення до їх наслідків. Враховуючи втрати, спричинені надзвичайними ситуаціями щороку, ефективне управління та боротьба з надзвичайними ситуаціями стало нагальною проблемою для сучасного світу.

Для проведення аналізу ризиків і вразливостей необхідно виконати наступні кроки:

- Визначити характер, масштаби і ступінь ризику виникнення небезпек;
- Визначити наявність і ступінь уразливості;
- Визначити наявні можливості та ресурси;
- Визначити прийнятні рівні ризику, витрат і вигод;
- Розробка методів встановлення пріоритетів щодо часу, ефективності розподілу ресурсів та результатів, для захисту людей і ключових ресурсів і зниження загальних втрат;
- Розробка ефективних та адекватних систем управління для впровадження та контролю.

Боротьба з надзвичайними ситуаціями зазвичай являє собою циклічний процес, що складається з наступних чотирьох основних етапів:

1. Підготовка — планування для різних катастроф, які можуть статися в цьому районі;
2. Реагування — видача попереджень до катастрофи і проведення початкових дій під час катастрофи;
3. Відновлення — проведення заходів щодо повернення постраждалого району та громади до нормального стану;
4. Пом'якшення наслідків — вжиття заходів, які зводять до мінімуму наслідки НС і запобігають повторення такого ж збитку.

Зазвичай кінець одного етапу є також початком наступного, але іноді етапи можуть перекриватися і кілька можуть відбуватися одночасно. Своєчасне прийняття рішень на кожному етапі призводить до підвищення готовності, більш раннім попередженням і зниження вразливості.

Повний цикл боротьби з надзвичайними ситуаціями також включає в себе розробку ефективних планів та державної політики щодо усунення причин стихійних лих та / або пом'якшення їх наслідків для майна, людей та інфраструктури.

Визначено наступні загальні вимоги до ефективної системи управління НС:

— Комунікаційні можливості. Як правило, в разі виникнення надзвичайної ситуації в службу екстреного реагування та екстрені служби надходить цілий потік телефонних дзвінків. У той же час сплеск повідомлень, пов'язаних з надзвичайною ситуацією, може бути розміщений на багатьох сайтах соціальних мереж. Ці телефонні дзвінки та повідомлення надзвичайно важливі для пом'якшення наслідків інциденту та рятування людей у надзвичайних ситуаціях. Отже, канали зв'язку системи попередження та ліквідації надзвичайних ситуацій повинні забезпечувати достатню міцність для передачі інформації між громадськістю та системами реагування на надзвичайні ситуації;

— Раннє виявлення і негайне реагування. Негайне реагування на надзвичайну ситуацію має вирішальне значення для скорочення кількості жертв і матеріального збитку. Раннє виявлення має вирішальне значення для виявлення та характеристики НС, оцінки та поширення інформації про можливі пошкодження, спричинені майбутнім інцидентом. Швидке реагування на надзвичайну ситуацію знижує вартість збитку, надаючи миттєві докази для командного центру, щоб своєчасно прийняти точне рішення;

— Ефективне управління інформацією. Оскільки основною метою управління надзвичайними ситуаціями є формування міцної основи для прийняття рішень в надзвичайних ситуаціях, воно повинно грамотно збирати, інтегрувати, організовувати, аналізувати і обмінюватися інформацією на всіх етапах надзвичайної ситуації;

— Розподіл ресурсів. У надзвичайних ситуаціях розумний розподіл ресурсів допомагає звести до мінімуму фізичні та соціальні наслідки, викликані інцидентом, враховуючи обмежені ресурси для порятунку та пом'якшення наслідків;

— Безпека та конфіденційність. Для ефективного здійснення плану запобігання надзвичайним ситуаціям та відновлення важливо гарантувати безпеку даних про надзвичайні ситуації, особливо у випадку катастрофи, такої як терористичний акт. Крім того, останнім часом все більше уваги приділяється проблемі конфіденційності. Система боротьби з надзвичайними ситуаціями повинна встановлювати належні

засоби авторизації доступу, щоб запобігти витоку приватної інформації в суспільне надбання для неправомірного використання.

2.2. Збір інформації

Обмін інформацією під час і після періодів НС може значно зменшити втрати, викликані катастрофою. Це пояснюється тим, що він дозволяє людям краще використовувати наявні ресурси, такі як матеріали для надання надзвичайної допомоги та медикаменти, і забезпечує канал, по якому можна оперативно доставляти повідомлення про жертви і втрати в кожному постраждалому районі. Управління надзвичайними ситуаціями на основі даних відноситься до застосування передових технологій збору та аналізу даних для досягнення більш ефективного і гнучкого управління надзвичайними ситуаціями [13].

Отримання даних з різних джерел стає все простіше і простіше в міру того, як уряди та організації виділяють все більше ресурсів на “звільнення” даних зі своїх закритих, застарілих і заблокованих баз даних.

Зрушення в мисленні перейшло від захисту всіх даних від сторонніх до визнання цінності певних загальних даних в різних організаціях. Тим не менш, в сховищах організацій все ще існує багато даних.

Якщо уряд не бажає і не готовий надати своєму громадянину вільний і нецензурний доступ до інформаційних каналів і засобів поширення, то недовіра і страх можуть перешкоджати участі в проектах створення карт НС, особливо якщо вони носять політичний характер.

Краудсорсинг, тобто залучення можливостей невизначеної кількості людей за допомогою інформаційних технологій з метою вирішення завдань, передбачає збір величезних обсягів даних з декількох джерел, які повинні бути оброблені з високою швидкістю. Ця робота зазвичай проводиться добровольцями та технічними спільнотами, які здатні швидко реагувати завдяки своїм децентралізованим глобальним, неієрархічним структурам [14]. Наприклад, Мур і Веріті відзначають, що

майже 230 000 записів у соціальній мережі Twitter (твітів) були зібрані і оброблені протягом двох днів після того, як тайфун Хайян обрушився на Філіппіни в 2013 році [15]. Це не призводить до помилкової думки, що деякі дані це добре, а чим їх ще більше тим краще, оскільки тільки 800 з цих твітів надали аварійним службам необхідні, дієві дані. Хоча це становить лише 0,35% інформації з мережі Twitter, зібраної натопом, швидке просіювання даних виділило ключові області руйнування та потенційно врятувало життя людей.

Незважаючи на обмежені практичні результати, отримані завдяки наведеному вище випадку, технології тепер дозволяють громадянам світу більш тісно взаємодіяти з подіями, які важливі для них як для окремих людей, без бар'єрів географічної дистанції. Це призводить до формування міжнародних мереж, об'єднаних спільними переконаннями і оснащених для участі в онлайн-колективних діях [16]. І хоча гуманітарні організації як і раніше є опорою кризових зусиль, вони більше не є єдиними учасниками гуманітарної сфери. Швидше за все, Інтернет надає населенню в цілому можливість самоорганізуватися і брати участь в якості добровольців, часто в рамках волонтерських та технічних спільнот [17].

2.2.1. Інформаційна потреба.

Люди є шукачами інформації, покладаючись в першу чергу на свої власні соціальні мережі [7]. Після катастрофи громадськість спочатку шукає найбільш поширені і звичні канали: телефонні дзвінки, електронну пошту або текстові повідомлення. У разі невдачі вони звертаються до альтернативних та / або офіційних джерел інформації [18].

Люди в надзвичайних ситуаціях використовують будь-які доступні засоби для пошуку інформації.

Люди шукають інформацію для:

- самих себе;
- щоб дізнатися про надзвичайну подію;
- того, щоб знайти їх сім'ю і друзів;
- зменшити невизначеність щодо того, що сталося.

Люди будуть шукати будь-яку доступну інформацію, включаючи газети, телебачення та інтернет [18].

Скіннер описує, як вона, як дослідник, збирає та узагальнює інформацію з різних джерел під час надзвичайних ситуацій і публікує її, щоб інформувати тих, хто постраждав від цієї події [19].

У випадках, що загрожують здоров'ю населення, велике значення має здатність діяти, особливо в тих випадках, коли очікується прийняття заходів або дій з виявлення ризику.

Повідомлення повинні поширюватися швидко і бути досить простими, щоб люди з високою тривожністю могли їх правильно зрозуміти [20].

З розвитком інтернет-технологій та інструментів користувачі в надзвичайних ситуаціях звертаються до Інтернету, щоб шукати інформацію, яка стосується їх регіону [3].

Ці нові можливості створили нову форму “громадянської журналістики”, яка дозволяє брати участь під час надзвичайних ситуацій. Інтернет стає все більш надійним інструментом, оскільки традиційні медіа-канали страждають від збоїв і збитків, викликаних надзвичайною ситуацією [21].

2.2.2. Об'єднання інформації з декількох джерел.

Під час надзвичайної ситуації інформація публікується так швидко, що люди, які реагують на неї, не можуть переглянути всі дані. Отже, для підсумовування необхідні алгоритми реального часу. Крім того, у міру розвитку надзвичайної ситуації опис ситуації також повинен еволюціонувати з плином часу.

Ситуаційна інформація вкрай важлива під час надзвичайних ситуацій, тому необхідно задіяти всі можливі джерела інформації. Звідси і численні джерела інформації, такі як новинні репортажі, соціальні мережі (Twitter, Facebook та ін.), SMS або WhatsApp повідомлення з мобільних телефонів і їх слід використовувати разом.

Проблема включення різних джерел інформації полягає в тому, що форма написання, яка використовується в різних джерелах, може бути різною. Наприклад, повідомлення новин пишуться офіційно, в той час як повідомлення в соціальних

мережах часто пишуться неофіційн. В той час як репортаж новин про землетрус в Непалі в 2015 році буде згадувати "Катманду", дослідники спостерігали, що кілька твітів використовують аббревіатуру "КТМ" для одного і того ж міста. Отже, інтелектуальні алгоритми необхідні для роботи з різним словниковим запасом різних джерел інформації. Рой робить перший крок у цьому напрямку, де нейромережева модель використовується для побудови загального простору вбудовування з різних словників різних джерел інформації Facebook, Twitter і WhatsApp [22].

2.2.3. Захист від дезінформації та інших видів шкідливого контенту.

Технологічна еволюція та недорогі інструменти дозволяють громадськості брати участь у процесі збору даних. Однак, оскільки кінцеві користувачі отриманих даних не знають про рівень точності пристроїв, або досвіду, або рівню підготовки волонтерів, необхідність оцінки якості даних перед використанням стає ще більш важливою.

Під час надзвичайних ситуацій, серед населення широко поширені паніка і напруженість. Не тільки жертви, а й волонтери залишаються в стані стресу, через який дезінформація і чутки здатні просочитися в мережу [24]. Виявити таку дезінформацію і чутки непросто, оскільки в такі моменти навіть дійсно відомі люди можуть мимоволі поширювати чутки. Насправді, об'єднання інформації з декількох джерел може бути хорошим способом виявлення дезінформації.

Ще одним видом шкідливого контенту, який часто публікується під час надзвичайних подій, є обцинний контент, орієнтований на певні релігійні або соціальні групи. Дивно, що такий контент публікується як під час соціально-політичних надзвичайних ситуацій (наприклад, терактів), так і під час природних лих (наприклад, повеней та землетрусів). Необхідно розробити методи виявлення такого контенту і подальшого ефективного поводження з ним. Наприклад, Рудра пропонує використовувати антиобцинний контент для протидії ефектам розміщення обцинного контенту [24].

2.3. Вимоги до розробки

У 2007 році Національна дослідницька рада (NRC) опублікувала книгу "Successful Response Starts with a Map: Improving Geospatial Support for Disaster Management", написану Комітетом з картографічної науки. У цій доповіді описується сучасний стан геопросторових даних та інструментів для управління надзвичайними ситуаціями та підкреслюється необхідність вдосконалення методів використання цих даних та інструментів. Механізми обміну даними, використання супутникових зображень та Інтернет-служб для надання даних відносяться до числа найважливіших вимог для використання геопросторових технологій [25].

2.3.1. Метод доступу.

Метод доступу відноситься до механізму, за допомогою якого здійснюється доступ до даних для інтеграції. Може виконуватись доступ через віддалені служби або інтерфейси, такі як веб-служби або служби баз даних і API, або прямий доступ до файлів через FTP, HTTP, RSS. Ці методи впливають на те, чи можуть дані запитуватися і взаємодіяти з іншими наборами даних, а також чи повинні вони зберігатися локально.

2.3.2. Модель даних.

Модель даних або схема відноситься до логічної внутрішньої організації елементів даних. Ця організація керує тим, як структуруються об'єкти та здійснюється доступ до них, включаючи вкладені файли, стовпці, рядки. Модель має велике значення, оскільки вона впливає на те, як будуються запити і чи доступні унікальні ідентифікатори для просторових об'єктів або подій.

Інформація, необхідна для реагування на надзвичайні ситуації, згрупована в два великих кластери: динамічна інформація (ситуаційна і оперативна) і статична (існуюча) інформація.

Дані, зібрані під час аварійного інциденту, позначаються як динамічні дані, в той час як інформація, що існує до аварії, називається статичною інформацією.

Прикладами динамічної інформації є:

- Інцидент: місце, природа, масштаб;
- Наслідки: уражена ділянка і розвиток в часі, можливість ескалації;
- Збитки: пошкоджені об'єкти, пошкоджена інфраструктура;
- Жертви: вбиті, поранені, зниклі безвісти, люди і тварини що потрапили в пастку;
- Доступність: входи в будівлі, в'їзні та виїзні маршрути, напрямок руху, заблоковані дороги
- Тимчасові центри: місця для розміщення людей (і тварин), центри допомоги, морги
- Знезараження: центри знезараження, транспортні засоби, будинки та інфраструктура для знезараження, люди і тварини для знезараження;

Динамічна інформація збирається з процесів одного кластера (тобто від суб'єктів, відповідальних за цей процес) і призначена для спільного використання з іншими кластерами/суб'єктами. Ця інформація може бути параметрами інциденту, такими як масштаб, розвиток, які регулярно оновлюються; кількість жертв з урахуванням різних категорій, таких як захоплені, поранені, легкопоранені, смерть, зниклі безвісти, які також регулярно оновлюються, або виміром.

У разі виявлення небезпечних речовин в повітрі, воді або в ґрунті, для збору проб направляються спеціальні вимірювальні групи — результати вимірювань повідомляються в командно-диспетчерський центр і аналізуються фахівцем.

2.4. Інтерактивні карти в Україні та світі

Цунамі в Індонезії та ураган "Катріна" свідчать про важливі досягнення, які були досягнуті завдяки використанню інтернету і геоінформаційних систем. Наприклад, катастрофа в Індонезії продемонструвала здатність швидко надавати дистанційно отримані дані як до, так і після події, що показують ступінь збитку. Провал урядових установ після урагану "Катріна" призвів до того, що багато людей самі проявили ініціативу та створили карти місць пожертувань і надання надзвичайної допомоги.

Зараз світ накопичив значний досвід в питанні створення систем моніторингу надзвичайних ситуацій не тільки природного походження, а і соціально-політичного характеру, але сама концепція, організаційна структура та технічна підтримка таких служб і систем постійно вдосконалюються.

Світовий досвід показує, що під час надзвичайних ситуацій спільні зусилля країн і веб-систем можуть підвищити продуктивність дій служб з порятунку для забезпечення своєчасного реагування та мінімізації збитку і втрат [26].

2.4.1. Українська платформа Liveuamap.

Нині одним з найуспішніших антикризових веб-картографічних проєктів в країні є «Liveuamap», який створений і підтримуваний Altwork IT на платформі Google Maps. Проєкт користується надзвичайною популярністю та, наприклад, в серпні 2014 року мав до 160 000 відвідувань на день [27].

На сайті проєкту можна ознайомитися не тільки з останніми даними про надзвичайні ситуації в Україні, а також в багатьох інших країнах. Сайт підтримує багато мов та зараз основна аудиторія проєкту припадає на західні країни, а українці складають лише 10% від загального числа користувачів.

Різні теми подій показані за допомогою іконок, що дозволяє швидко розрізняти характери надзвичайних ситуацій (рисунок 2.1).

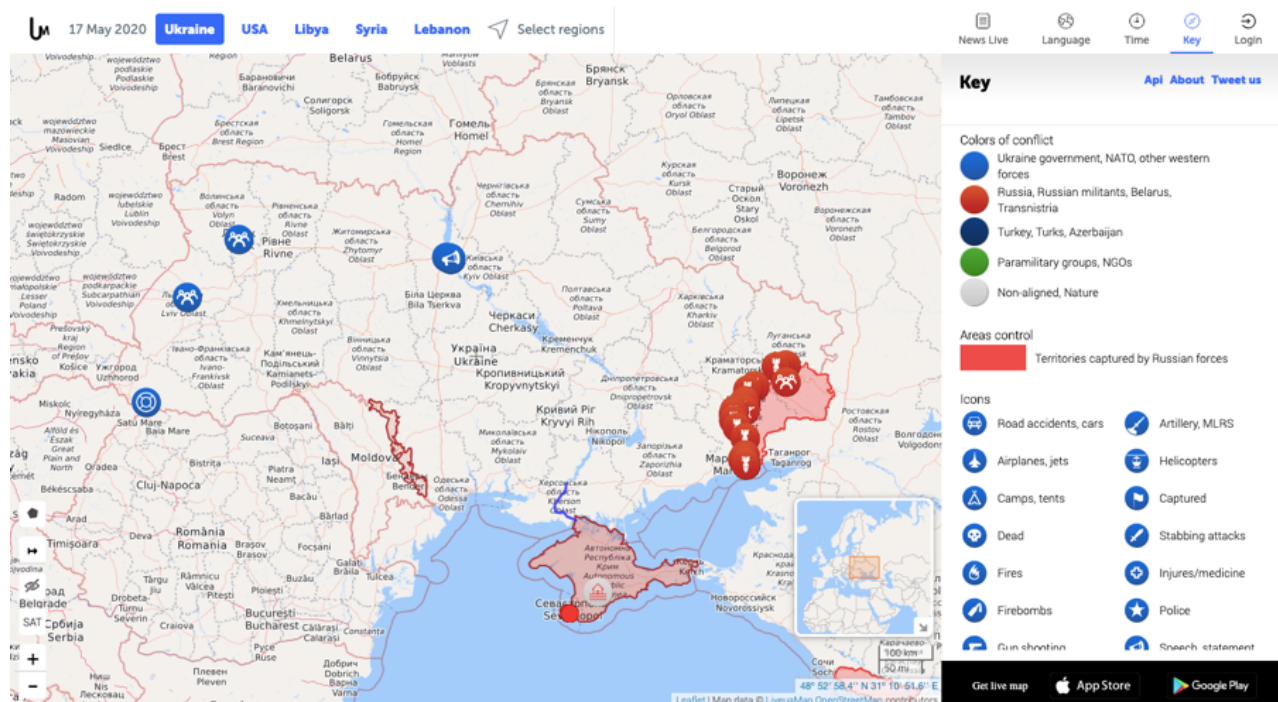


Рисунок 2.1 — Темі подій

Окрім веб-сайту, система також існує у вигляді мобільного додатку як для телефонів на базі мобільної операційної системи iOS, так і на базі Android.

2.4.2. Угорська система RSOE — Emergency and Disaster Information Service.

Угорська національна асоціація **The National Association of Radio Distress-Signalling and Infocommunications (RSOE)** керує службою інформації про надзвичайні ситуації та катастрофи (**EDIS**) в рамках свого власного веб-сайту, який має на меті відстежувати і документувати всі події на Землі, які можуть викликати надзвичайні ситуації [28].

Громадяни можуть стежити за катастрофами, які сталися в Угорщині, в Європі або в світі на карті надзвичайних ситуацій безперервно з періодом оновлення інформації до 15 хвилин (рисунок 2.2).

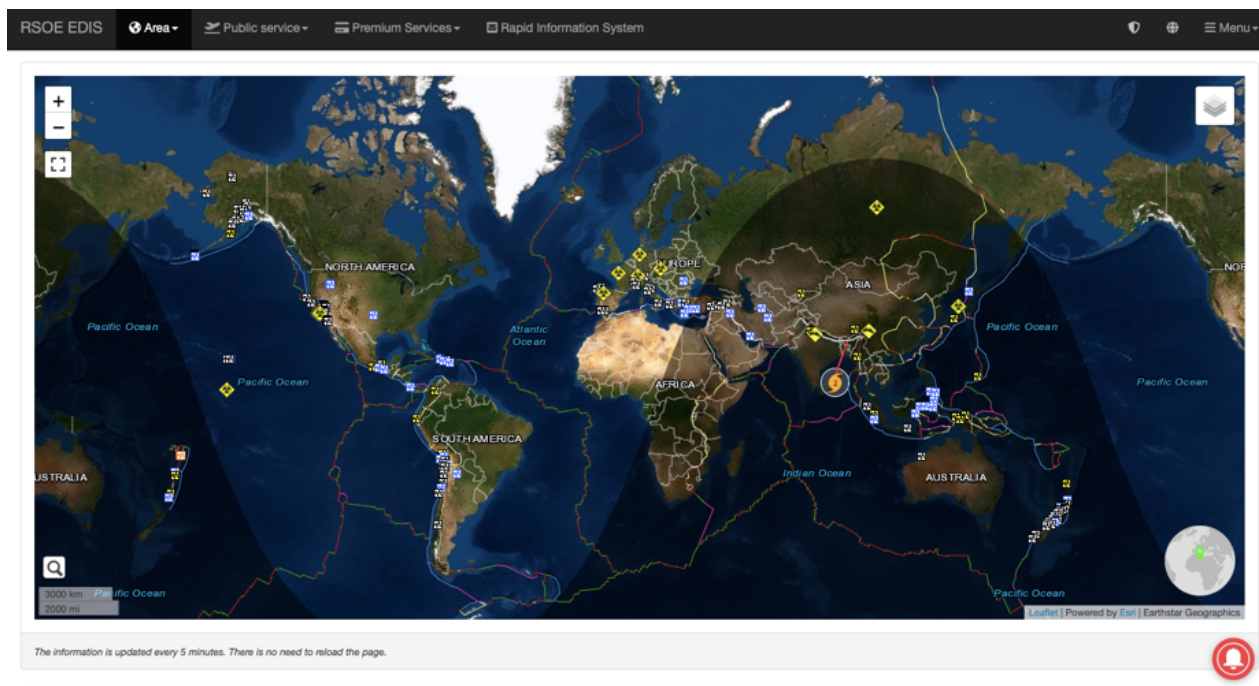


Рисунок 2.2 — Карта надзвичайних ситуацій EDIS

Сервіс відстежує та обробляє дані кількох світових організацій, щоб отримати швидку та сертифіковану інформацію.

Система перевіряє інформацію, що надходить, і зберігає її в центральній базі даних, відсортованій за категоріями. Після обробки інформації вона негайно відправляє її електронною поштою (або в іншому форматі) для організацій та осіб, які її запросили. Система прагне до того, щоб оброблені дані були достовірними, щоб уникнути можливої панічної ситуації, викликаной неправдивою інформацією. Саме тому сервіс намагається створювати і підтримувати контакт з усіма організаціями, які можуть надати достовірну інформацію.

Сервіс також має додаток для Android в Google play. Нові події з'являються не тільки на сайті EDIS, але і в додатку для смартфонів RSOE EDIS Notifier. Використовуючи технологію PUSH, додаток дає вам інформацію в реальному часі про події в світі.

2.4.3. Copernicus Emergency Management Service.

Служба управління надзвичайними ситуаціями Copernicus (EMS) надає інформацію для реагування на надзвичайні ситуації у зв'язку з різними видами лих,

включаючи метеорологічні небезпеки, геофізичні небезпеки, навмисні і випадкові техногенні катастрофи та інші катастрофи, а також заходи щодо попередження, забезпечення готовності, реагування та відновлення [29].

Останні події можна побачити на карті, що знаходиться на головній сторінці веб-системи EMS (рисунок 2.3).

Copernicus Emergency Management Service - Mapping

A service in support of European emergency response

Map above displays only latest Copernicus EMS - Mapping Activations. To see a Map of All Activations, go to section Map of Activations in Rapid Mapping or in Risk and Recovery Mapping sub-menus respectively.

Act. Code	Title	Act. Date	Type	Country/Terr.	Feed
EMSR438	Flood in Uganda	2020-05-15	Flood	Uganda	
EMSR437	Flood in South West of France	2020-05-10	Flood	France	
EMSN076	Germany BEAM update	2020-04-24	Other	Germany	
EMSN075	Forest fires preparedness, Ukraine	2020-04-24	Wildfire	Ukraine	
EMSR436	Fire in Podlaskie Voivodeship, Poland	2020-04-22	Wildfire	Poland	
EMSR435	Fire in Chernobyl, Ukraine	2020-04-14	Wildfire	Ukraine	
EMSN071	Wildfires, Preparedness, Arnsberg, Germany	2020-04-09	Other	Germany	
EMSR434	Tropical Cyclone HAROLD in Vanuatu	2020-04-06	Storm	Vanuatu	
EMSD433	COVID-19 outbreak and spread in Italy	2020-04-06	Epidemic	Italy	

Рисунок 2.3 — Головна сторінка EMS

Користувачі — це суб'єкти та організації на регіональному, національному, європейському та міжнародному рівнях, що діють в галузі управління кризами в державах-членах ЄС, держави-учасники Європейського механізму цивільного захисту, генеральні директори комісії (DGs) і Агентства ЄС, Європейська служба зовнішніх дій (EEAS), а також міжнародні організації з надання гуманітарної допомоги.

Існує три різних категорії користувачів веб-системи:

1. Авторизовані користувачі, які можуть запусити сервіс, відправивши форму запиту на обслуговування (SRF) безпосередньо в Європейський Координаційний

Центр реагування (ERCC). До числа уповноважених користувачів відносяться національні координаційні центри в державах-членах ЄС і в більшості країн, що беруть участь в Європейському механізмі цивільного захисту, а також служби ЄС, ситуаційний центр CEAC і делегації ЄС.

2. Асоційовані користувачі, які включають місцеві, регіональні та інші державні структури; міжнародні урядові організації (наприклад, агентства ООН, Світовий банк) та національні та міжнародні неурядові організації; INTCEN, супутниковий центр ЄС.

3. Звичайні публічні користувачі не мають права запускати сервіс, але можуть бути проінформовані про запит активації через веб-портал.

2.4.4. PDC DisasterAWARE.

DisasterAWARE — це інтегрована веб-платформа від Pacific Disaster Center (PDC), що забезпечує ситуаційну обізнаність, підтримку прийняття рішень та обмін інформацією для управління даними НС. Оповіщення про інформацію активної небезпеки засновані на даних надійних попереджувальних агентств (відповідно до стандартів PDC) [30].

НС можна побачити на карті, що знаходиться на головній сторінці веб-платформи DisasterAWARE (рисунок 2.4).

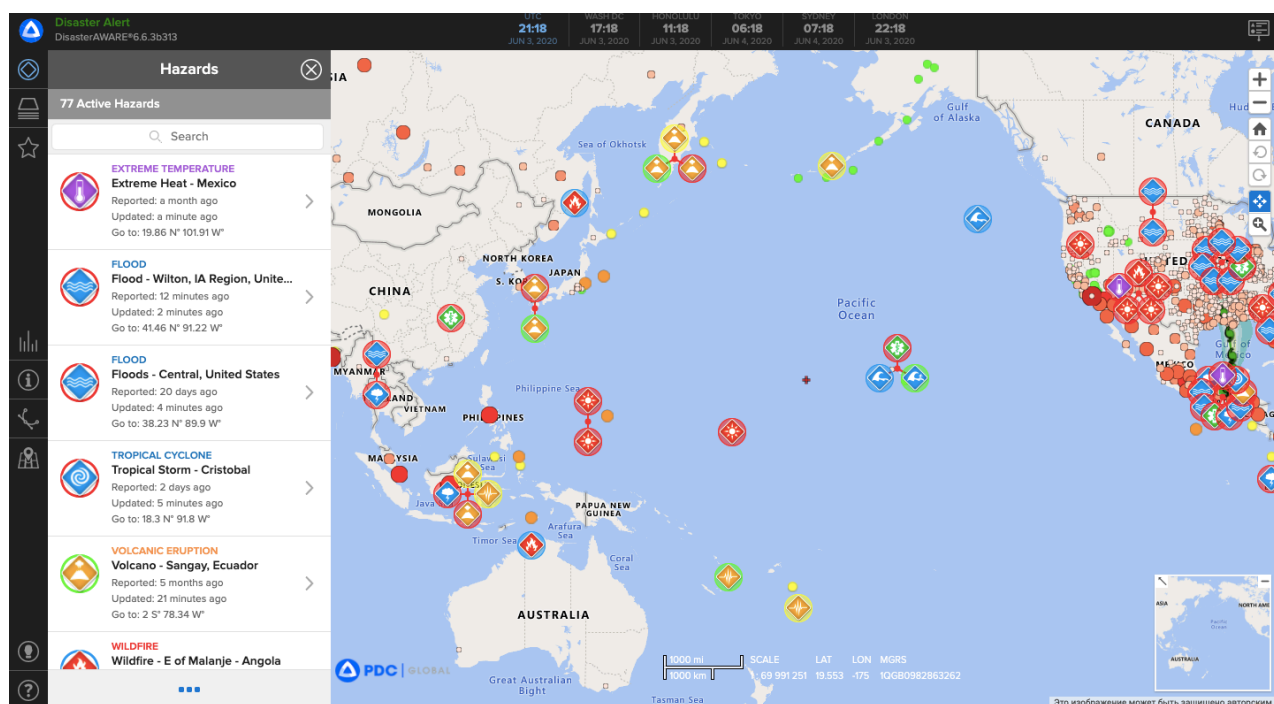


Рисунок 2.4 — Головна сторінка DisasterAWARE

Веб-платформа DisasterAWARE надає:

- Попередження про всі НС;
- Аналіз і оцінку ризиків;
- Візуалізацію, що дозволяє вимірювати, малювати полігони, розміщувати маркери;
- Шляхи шторму, швидкість вітру, накопичення опадів, Дартмутський метод величини повені, глобальний шар хмар, температура поверхневого повітря, температура поверхні моря;
- НС в межах регіонів, що цікавлять нас;
- Регіональний огляд чисельності і щільності населення;
- Можливість уряду або міжнародних організацій створити обліковий запис для розширеного доступу.

3. ЗАСОБИ РОЗРОБКИ

Для розробки веб-системи було використано багато технологій, щоб забезпечити:

- розробку інтерфейсу;
- створення бази даних та керування нею;
- візуалізацію надзвичайних ситуацій на карті;
- можливість автентифікації користувачів;

3.1. Середовище розробки Visual Studio / VS Code

У своїй основі Visual Studio Code є безкоштовною середовищем розробки. VS Code допомагає слідкувати за правильністю написання коду через підсвічування синтаксису, зіставлення дужок, автоматичним відступом, виділенням полів та фрагментів [31]. Інтуїтивно зрозумілі комбінації клавіш та просте налаштування дозволяють легко орієнтуватися в коді.

Visual Studio Code включає в себе інтерактивний налагоджувач, що дозволяє перевіряти змінні, переглядати стеки викликів і виконувати команди в консолі.

VS Code також має інструменти для виконання спільних завдань. VS Code підтримує Git для того, щоб працювати з системою управління версіями, не виходячи з редактора.

VS Code має вбудовану підтримку Node.js розробки з використанням JavaScript і TypeScript. VS Code також включає в себе інструменти для веб-технологій, таких як HTML, CSS, SCSS і JSON.

Через VS Code, за допомогою Microsoft Azure, з'являється можливість розмістити свої Angular сайти.

3.2. Мова програмування C#

C# — об'єктно-орієнтована мова програмування.

Програми на мові C# працюють на платформі .NET Framework — невід'ємному компоненті Windows [32], який включає в себе віртуальну виконавчу систему, звану *common language runtime (CLR)*, і уніфікований набір бібліотек класів. CLR — це комерційна реалізація корпорацією Майкрософт інфраструктури спільної мови (CLI), міжнародного стандарту, який є основою для створення середовищ виконання і розробки, в яких мови і бібліотеки працюють разом без проблем.

3.3. Середовище CLR

Платформа .NET Framework надає середовище виконання, що називається *common language runtime (CLR)*, яке запускає код і надає служби, що полегшують процес розробки [33].

Компілятори та інструменти надають функціональні можливості середовища *common language runtime*. Код, що розробляється за допомогою мовного компілятора, призначеного для середовища виконання, називається керованим кодом; він використовує такі функції, як міжмовна інтеграція, обробка міжмовних винятків, підвищена безпека, підтримка версій, спрощена модель взаємодії компонентів, а також служби налагодження.

Середовище допомагає спростити процес розробки, надаючи різні служби. В принципі, воно відповідає за управління виконанням програм .NET незалежно від будь-якої .NET мови програмування. Внутрішньо CLR реалізує VES (Virtual Execution System), яка визначена в реалізації Microsoft CLI (Common Language Infrastructure).

3.4. Платформа .NET Core

.NET Core — це безкоштовна платформа з відкритим кодом для створення хмарних додатків, таких як веб-додатки та додатки Інтернету речей [32]. Вона призначена для роботи як в хмарі, так і в локальному середовищі.

3.5. Служба Azure Web Apps

Служба додатків Azure — це служба на основі HTTP для розміщення веб-додатків. Вона надає можливість розробляти на .Net, .Net Core, Node.js та багато інших [34]. Програми легко запускаються на будь-якій платформі як в середовищі Windows, так і в середовищі Linux.

3.6. SQL Server (SQL Azure)

Microsoft Azure дозволяє створювати бази даних SQL Server за допомогою моделі database as a Service (DBaaS), тому не потрібно турбуватися про керування віртуальними машинами або.

База даних SQL Azure — це повністю керована платформа як послуга (PaaS) Database Engine, яка обробляє більшість функцій управління базами даних, таких як оновлення, виправлення, резервне копіювання та моніторинг без участі користувача [35]. База даних SQL Azure завжди працює на останній стабільній версії компонента SQL Server Database Engine. Можливості PaaS, вбудовані в базу даних SQL Azure, дозволяють зосередитися на діях з адміністрування та оптимізації баз даних для конкретного домену.

За допомогою бази даних SQL Azure можна створити доступний і високопродуктивний рівень зберігання даних для додатків і рішень в Azure. База даних SQL є гарним вибором для різних сучасних хмарних додатків, оскільки вона дозволяє обробляти як реляційні дані, так і нереляційні структури, такі як графіки, JSON та XML.

3.7. JWT автентифікація

JSON Web Token (JWT) — це технологія для автентифікації [36]. Після входу користувача в додаток, додаток створить JWT маркер і відправить його назад користувачеві. Сервер буде знати, що це той самий конкретний користувач, тому що запит підписаний з його унікальним ідентифікатором. Наступні запити користувача включатимуть JWT маркер. Маркер повідомляє серверу, до яких маршрутів, служб і ресурсів користувачеві дозволений доступ. JWT можна легко використовувати в декількох доменах, тому вони часто використовуються для єдиного входу.

JWT — це механізм перевірки власника деяких даних JSON. Це закодований рядок, який може містити необмежену кількість даних (на відміну від файлів cookie).

Коли сервер отримує JWT, він може гарантувати, що даним, які в ньому знаходяться, можна довіряти. Жоден посередник не може змінити JWT після його відправки.

3.8. JavaScript

JavaScript — це мова програмування, яка дозволяє реалізувати складні функції на веб-сторінках та використовується для створення та керування динамічним вмістом веб-сайту, тобто всім, що рухається, оновлюється або іншим чином змінюється на екрані, не вимагаючи ручного перезавантаження веб-сторінки. За допомогою цієї мови стає можливим розміщення слайд-шоу з фотографіями, пропозицій щодо заповнення тексту у формах, відображення своєчасних оновлень вмісту сторінок, інтерактивних карт, анімованої 2D/3D графіки тощо.

У браузері JavaScript може виконувати все, що пов'язано з маніпуляцією веб-сторінкою, взаємодією з користувачем і веб-сервером.

3.9. Фреймворк Angular

AngularJS — це структурний фреймворк для динамічних веб-додатків. Він дозволяє використовувати HTML як шаблонну мову і дозволяє розширити синтаксис HTML для чіткого і лаконічного вираження компонентів програми. Тобто, AngularJS змушує робити браузер усе те що потрібно.

AngularJS — це JavaScript-фреймворк. Основна ідея Angular полягає в тому, щоб об'єднати добре відомі компоненти, шаблони і методи розробки в одному фреймворку, який простий у використанні і заохочує слідування кращим практикам веб-розробки [37]. Основна мета полягає в тому, що за допомогою Angular розробники можуть і будуть вибирати не просто будь-яке рішення, яке працює, а найвідоміше рішення — “Angular спосіб”.

Для досягнення цієї мети AngularJS підтримує стандартні компоненти, такі як моделі, представлення, контролери та служби. Це призводить до того, що Angular використовує два важливі шаблони проектування програмного забезпечення: контролер представлення моделі та ін'єкцію залежностей. Крім того, Angular використовує безліч абстракцій, звертаючи увагу на балансування модульності і складності.

Важливо відзначити, що AngularJS створений з особливою увагою до тестування. Він підтримує написання тестового коду, що на практиці призводить до більш високої якості програмного забезпечення і меншої кількості перерв в обслуговуванні. Angular також добре підтримує використання бібліотеки jQuery, що гарантує, що багато популярних методів веб-розробки можуть бути використані, як і раніше.

3.10. Google Maps API

Ми живемо в епоху Google Maps. Завдяки Google Maps API ми можемо вставляти геопросторові дані Google Maps на розроблюваний веб-сайт [38].

Оверлеї (накладення) — це об'єкти на карті, прив'язані до координат широти / довготи.

Щоб відзначити одне місце на карті, Google Maps надає маркери.

Маркер — тип накладення поодиноких місць на карті. Маркер визначає місце розташування на карті. За замовчуванням маркер використовує стандартне зображення (рисунок 3.1).

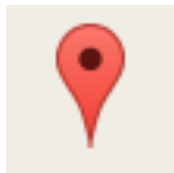


Рисунок 3.1 — Стандартний символ маркера

Маркери можуть відображати власні зображення, і в цьому випадку їх зазвичай називають "іконками". Маркери та іконки — це об'єкти типу маркер. Ми можемо встановити користувальницький значок в конструкторі маркера або викликати `setIcon()` на маркері.

Конструктор маркерів створює маркер. Для відображення маркера необхідно задати властивість `position`.

Маркери призначені для того, щоб бути інтерактивними. Наприклад, за замовчуванням вони отримують події натискання на кнопку миші, тому ми можемо додати слухач подій, щоб викликати інформаційне вікно, де буде відображена потрібна інформація. Також ми можемо дозволити користувачам переміщати маркер на карті, встановивши для властивості маркера, що перетягується, значення `true`.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Для створення веб-системи обліку надзвичайних ситуацій соціально-політичного характеру потрібне створення моделей бази даних, проектування архітектури системи та реалізацію серверної частини.

4.1. Опис функціональності системи

Для моделювання системи найбільш важливим аспектом є фіксація динамічної поведінки. Динамічна поведінка означає поведінку системи, коли вона працює.

Діаграми прецедентів (діаграми варіантів використання) складаються з акторів, прецедентів (варіантів використання) та їх взаємозв'язків. Ці діаграми використовуються для моделювання системи / підсистеми програми.

Діаграма прецедентів — це динамічна діаграма або діаграма поведінки в мові UML (Unified Modeling Language). Діаграми прецедентів моделюють функціональність системи за допомогою використання акторів і прецедентів. Прецеденти — це набори дій, служб і функцій, які повинна виконувати система. У цьому контексті “система” — це те, що розробляється або використовується, наприклад веб-сайт. “Актори” — це люди або сутності, що задіяні в рамках певних ролей в системі.

Деякі прецеденти пов'язані між собою відношеннями включення або відношеннями розширення.

Відношення включення (include) відображає такий зв'язок, коли один прецедент в момент виконання повністю включає інший.

Відношення розширення (extend) відображає зв'язок, коли один прецедент за певної умови повністю використовує (розширює) інший.

Нижче наведено діаграму прецедентів (рисунок 4.1), що представляє систему обліку надзвичайних ситуацій.

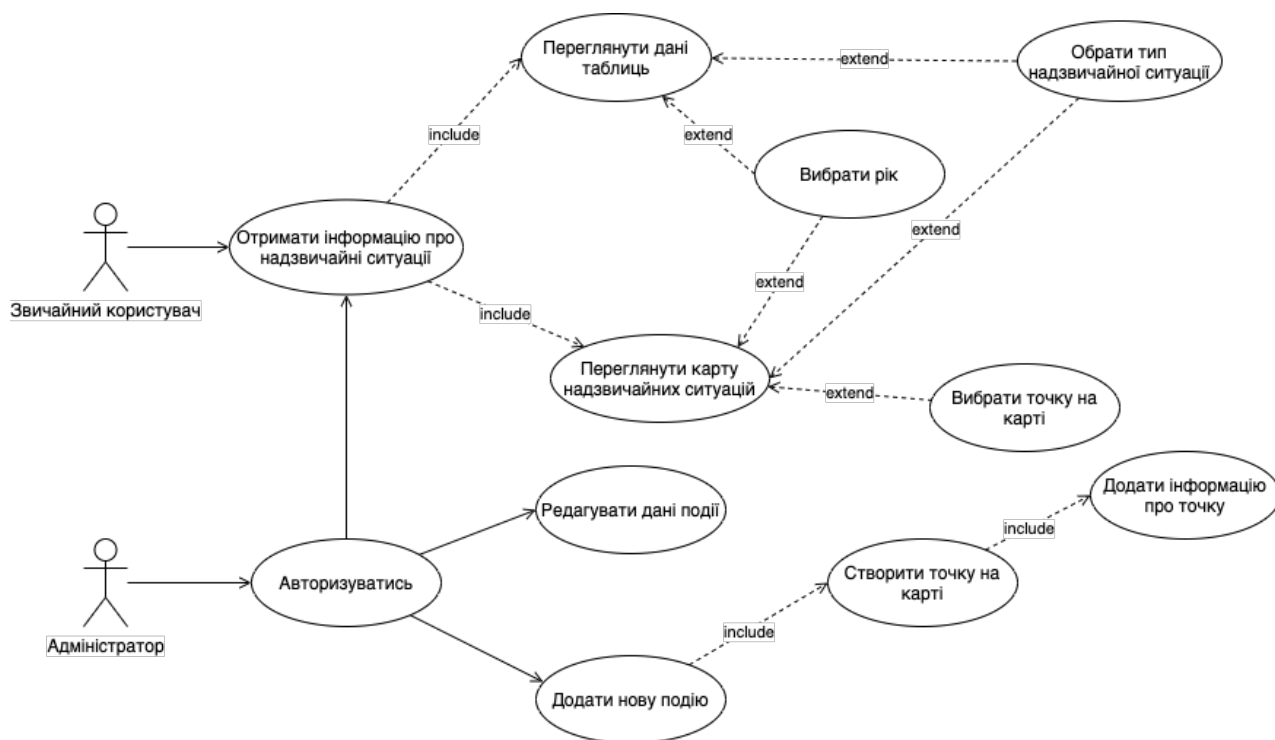


Рисунок 4.1 — Діаграма прецедентів системи

Отже, якщо ми подивимося на діаграму, то ми знайдемо одинадцять прецедентів, які зображені за допомогою еліпсів, та двох акторів, позначених символом людини.

Акторами даної системи є звичайний користувач та адміністратор.

Звичайний користувач має змогу отримати інформацію про надзвичайні ситуації, що включає в себе перегляд даних таблиць та перегляд карт надзвичайних ситуацій завдяки відношенню включення.

Прецеденти “Обрати тип надзвичайної ситуації” та “Вибрати рік” розширюють прецедент “Переглянути дані таблиць”, тобто вони надають змогу фільтрувати дані за типом чи роком надзвичайної ситуації.

Аналогічно прецедент “Переглянути карту надзвичайних ситуацій” розширюється від прецедентів “Вибрати рік”, “Обрати тип надзвичайної ситуації” та “Вибрати точку на карті”.

Адміністратор обов’язково має авторизуватись для подальшої роботи з системою. Йому, як і звичайному користувачу, надається змога отримати інформацію про надзвичайні ситуації.

Також, на відміну від звичайного користувача, адміністратор може додати нову подію та редагувати дані подій.

Прецедент “Додати нову подію” включає в себе “Створити точку на карті”, який у свою чергу включає “Додати інформацію про точку”.

4.2. Модель бази даних

Бази даних сьогодні входять до числа найбільш важливих структурних елементів Всесвітньої мережі. Знаходячись в основі більшості контент-орієнтованих веб-сайтів і додатків, бази даних виконують особливу місію — забезпечити добре організований механізм керування даними. Бази даних у розробці веб-сайтів / додатків тепер управляють мережею, пропонуючи швидкий і автоматизований спосіб зберігання, управління, видалення або вилучення інформації. Потужний набір можливостей баз даних визначив появу динамічних веб-сайтів, що відкрило нову сторінку в історії еволюції Інтернету.

База даних — це організована сукупність різних форм даних. БД також відомі як структуровані набори даних, які доступні у багатьох формах через комп'ютер.

Нам потрібні бази даних для:

— Управління великими фрагментами даних: ми можемо зберігати дані в звичайній електронній таблиці, але якщо ми додаємо великі фрагменти даних в лист, це просто не буде працювати. Наприклад: якщо наш розмір даних збільшується до тисячі записів, це просто приведе до проблем швидкості;

— Точності: при введенні даних в електронну таблицю стає важко управляти точністю, так як в таблиці немає ніяких перевірок;

— Простоти оновлення даних: за допомогою бази даних ми можемо швидко оновлювати дані. Крім того, кілька людей можуть одночасно редагувати дані;

— Безпеки даних: не можна заперечувати той факт, що наші дані менш захищені в електронних таблицях. Будь-який охочий може легко отримати доступ до файлу і внести в нього зміни. З базами даних у нас є групи безпеки, які ми встановлюємо для обмеження доступу;

— Цілісності даних: цілісність даних також стає важливим питанням при зберіганні даних в електронних таблицях. У базах даних ми можемо бути впевнені в точності і узгодженості даних завдяки вбудованим перевіркам цілісності і контролю доступу;

База даних розробленої системи містить в собі три таблиці щодо надзвичайних ситуацій (рисунок 4.2).

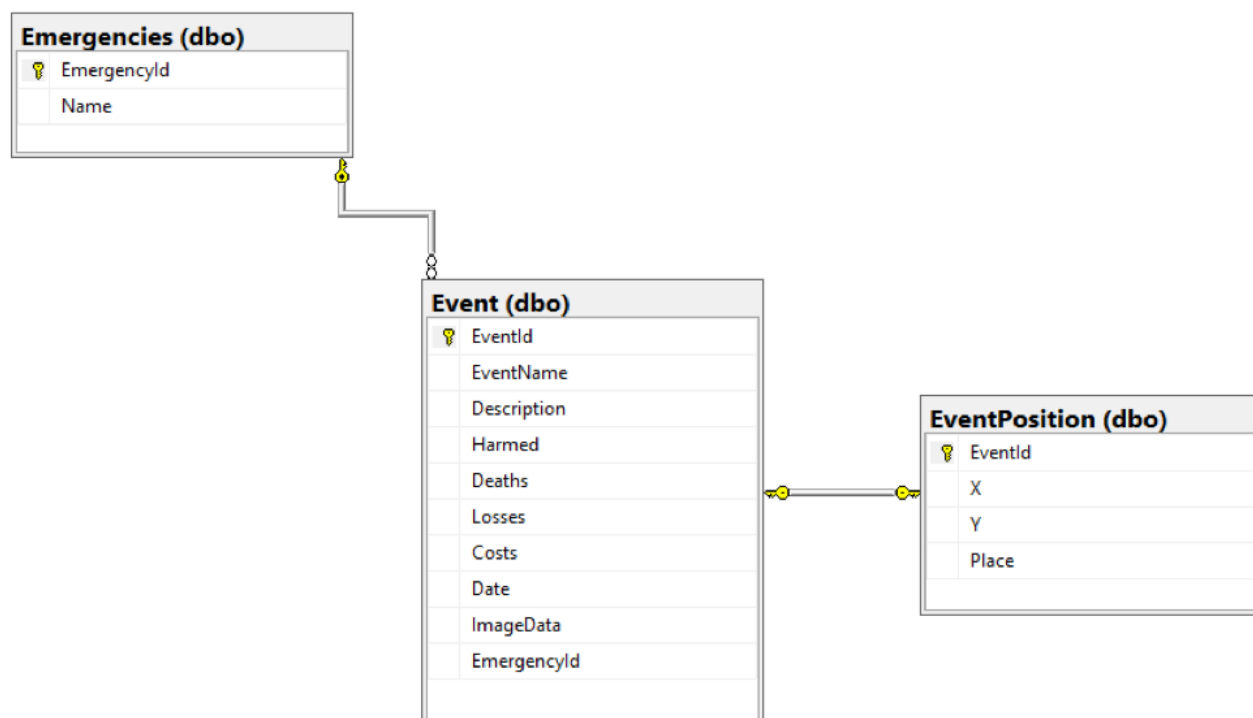


Рисунок 4.2 — База даних надзвичайних ситуацій

Перша таблиця складається з інформації про характер надзвичайної ситуації.

У другій таблиці знаходяться дані про саму подію (код ситуації, її назву, опис події, кількість постраждалих, число загиблих, дату тощо).

Третя таблиця містить дані про місце події та її координати на карті.

Для забезпечення різних типів користувачів системи, в базі також присутні таблиці, що надаються системою .NET Core Identity (рисунок 4.3).

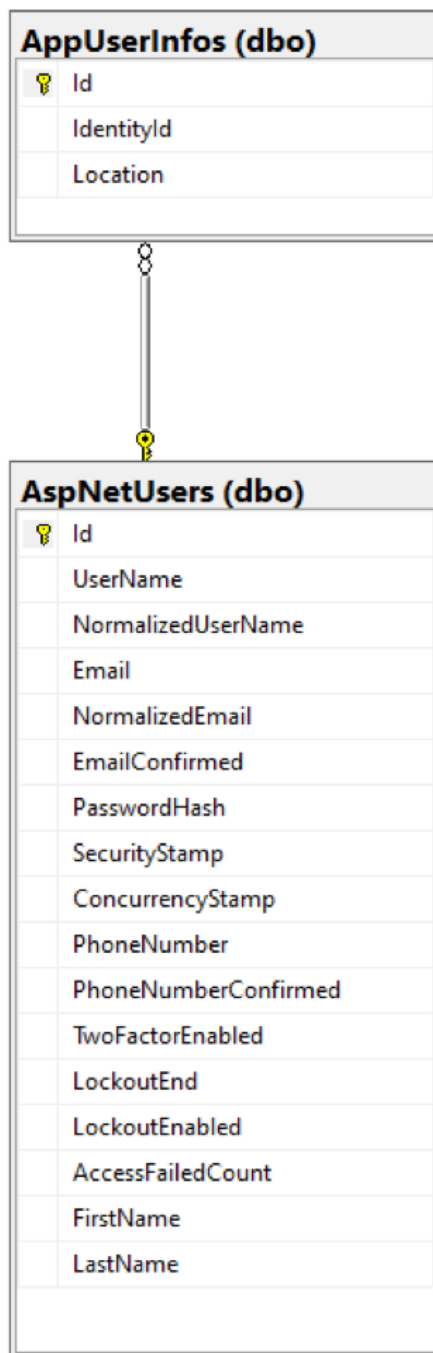


Рисунок 4.3 — База даних користувачів системи

Для авторизації користувачів використовується ASP.NET Core Identity, який підтримує функцію входу в інтерфейс користувача (UI) та керує користувачами, паролями, маркерами, підтвердженням електронної пошти.

Ідентифікація користувача налаштована за допомогою бази даних SQL Server, що розміщується в сховищі Azure.

4.3. Проектування архітектури системи

Для роботи з базами даних та для керування таблицями використовується Entity Framework Core. Entity Framework Core є ORM-інструментом (object-relational mapping — відображення даних на реальні об'єкти).

Для створення системи використовується підхід Model First. Завдяки даному підходу ми зможемо створити модель бази даних і діаграму класів в цілому за допомогою інструменту візуального проектування. Спочатку робиться модель, а потім по ній створюється база даних.

За предметною областю були сформовані відповідні моделі (представлені у вигляді класів). Потім моделі за допомогою механізму міграції були трансформовані в таблиці бази даних.

Кожен раз, коли база даних змінюється, модель може бути оновлена відповідним чином, без втрати даних.

Для створення системи був застосований шаблон проектування MVVM (Model-View-ViewModel). Схема роботи даного шаблону зображена на рисунку 4.4.

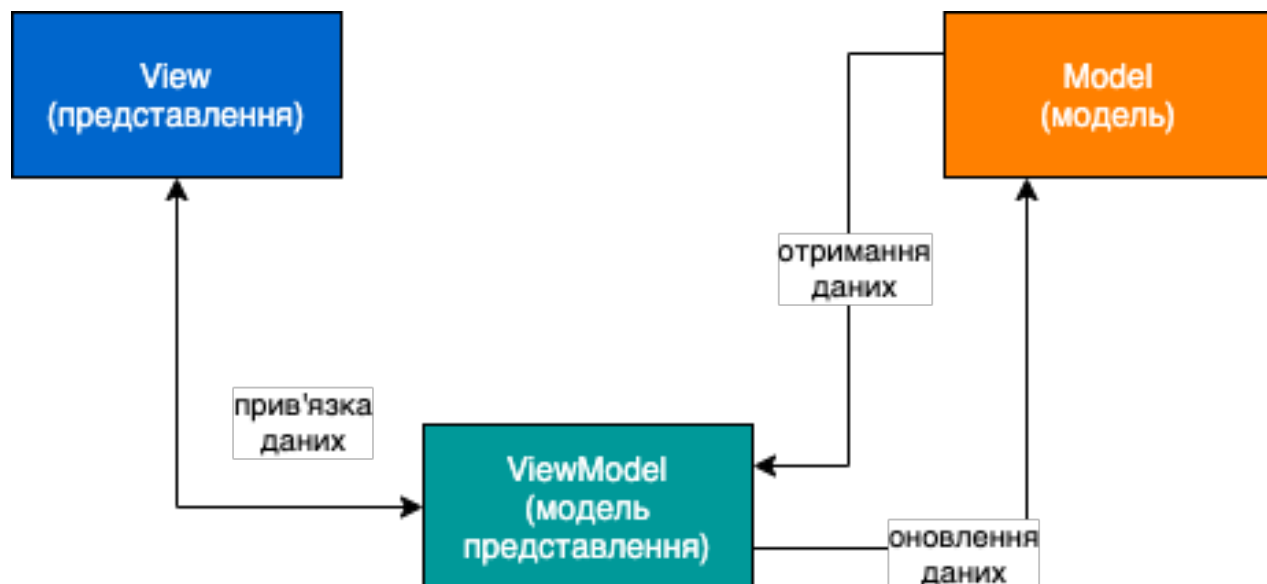


Рисунок 4.4 — Схема роботи MVVM шаблону

Шаблон проектування — це архітектура, яка описує яким чином вирішуються задачі, які часто зустрічаються при розробці програмних систем.

Model-View-ViewModel (MVVM) — це архітектурний шаблон, який полегшує відокремлення розробки графічного інтерфейсу користувача (представлення) від розробки бізнес-логіки або внутрішньої логіки (моделі), так що представлення не залежить від будь-якої конкретної платформи моделі.

Три основні компоненти MVVM:

- Модель (англ. Model);
- Представлення (англ. View);
- Модель Представлення (англ. ViewModel).

Модель являє собою фактичні дані та / або інформацію, з якими ми маємо справу. Прикладом моделі може бути контакт, що містить ім'я, номер телефону, адресу тощо.

Модель являє собою логіку роботи з даними і опис фундаментальних даних, необхідних для роботи програми.

У широкому сенсі ця модель забезпечує доступ до даних і служб, які необхідні нашому додатку. Саме тут відбувається справжня робота. У той час як модель представлення пов'язана з об'єднанням даних моделі, класи моделей виконують фактичну роботу програми.

Важливо запам'ятати, що модель містить інформацію, але не поведінку або сервіси, які керують цією інформацією. Модель не відповідає за форматування тексту для того, щоб виглядати красиво на екрані, або витяг списку елементів з віддаленого сервера (насправді, в цьому списку кожен елемент, швидше за все, буде являти собою власну модель). Бізнес-логіка зазвичай зберігається окремо від моделі і інкапсулюється в інші класи, які діють на модель.

Модель описує використовувани в додатку дані. Моделі можуть містити логіку, безпосередньо пов'язану цими даними, наприклад, логіку валідації властивостей моделі. У той же час модель не повинна містити ніякої логіки, пов'язаної з відображенням даних і взаємодією з візуальними елементами управління.

Представлення — це те, що більшості з нас знайоме, і єдине, з чим дійсно взаємодіє кінцевий користувач. Це представлення даних. Представлення вимагає певні бібліотеки, щоб зробити ці дані більш презентабельними.

Представлення — графічний інтерфейс (кнопки, вікна, списки тощо). Виступає спостерігачем події зміни значень властивостей або команд, що надаються моделлю представлення. У разі, якщо в моделі представлення змінилась якась властивість, то вона сповіщає всіх спостерігачів про це, і представлення, в свою чергу, запитує оновлене значення властивості з моделі представлення. У випадку, коли користувач впливає на який-небудь елемент інтерфейсу, представлення викликає відповідну команду, надану моделлю представлення.

Представлення визначає графічний інтерфейс, через який користувач взаємодіє з додатком.

Представлення — це структура, макет і зовнішній вигляд того, що користувач бачить на екрані. Представлення відображає модель та отримує взаємодію користувача з представленням (клацання на кнопку миші, ввід з клавіатури, жести тощо), і воно перенаправляє обробку цих даних у модель представлення через прив'язку даних (властивості, зворотні виклики подій), які визначаються для зв'язку представлення та моделі представлення.

Представлення також може мати поведінки, пов'язані з, наприклад, прийняттям користувальницького введення. Представлення управляє введенням даних (натисканням клавіш, рухом миші, жестами), що в результаті управляє властивостями моделі.

Модель Представлення пов'язує модель і представлення через механізм прив'язки даних.

Модель Представлення, з одного боку, абстракція представлення, а з іншого — обгортка даних з моделі, що підлягають зв'язуванню. Тобто, модель представлення містить модель, перетворену до представлення, а також команди, якими може користуватися представлення, щоб впливати на модель.

Модель Представлення також містить логіку з отримання даних з моделі, які потім передаються в представлення. Також модель представлення визначає логіку щодо оновлення даних у моделі.

Ми можемо описати Модель Представлення як стан даних у моделі.

Модель Представлення також надає методи та команди, які допомагають підтримувати стан представлення, керувати моделлю в результаті дій над представленням та ініціювати події в самому представленні.

Переваги використання архітектурного шаблону MVVM:

— Спільна робота: Відокремлюючи візуальну частину програми (користувальницький інтерфейс або графічний інтерфейс) від пов'язаного коду, можна надати одночасний доступ фахівцям з кожної області працювати над пов'язаними елементами. Дизайнери можуть працювати над користувальницьким інтерфейсом одночасно з розробниками, які працюють над кодом, не вимагаючи, щоб вони обидва працювали над одними і тими ж файлами одночасно;

— Простота тестування: Одним з важливих зауважень до програм з графічними інтерфейсом є те, що їх важко протестувати. Шаблон поділу, такий як MVVM, повинен розірвати зв'язок між логікою програми та користувальницьким інтерфейсом і таким чином зробити тестування більш доступним. Не повинно бути необхідності робити все тестування через користувальницький інтерфейс, і тому тести стають швидшими і простішими в налаштуванні та запуску. MVVM розриває зв'язок між логікою програми та інтерфейсом користувача та робить тестування більш доступним. З MVVM кожен фрагмент коду більш деталізований, і якщо він реалізований правильно, зовнішні і внутрішні залежності знаходяться в окремих фрагментах коду від частин з основною логікою, які потрібно перевірити. Це значно полегшує написання модульних тестів основної логіки;

— Простота обслуговування: Завдяки поділу між різними частинами коду програми, забезпечується певний рівень структури коду;

— Прозора комунікація: Модель представлення надає прозорий інтерфейс контролеру представлення, який він використовує для заповнення шару

представлення і взаємодії з шаром моделі. Це призводить до прозорої взаємодії між рівнями програми.

Отже, застосування шаблону MVVM надає функціональний поділ програми на три компоненти, які простіше розробляти і тестувати, а також надалі модифікувати і підтримувати.

4.4. Реалізація серверної частини системи

Серверна частина системи створена на .NET Core.

Система авторизації користувачів реалізована за допомогою JSON Web Token (JWT).

Автентифікація — це набір дій, які ми використовуємо для перевірки введених даних користувача в порівнянні з тими, що знаходяться в базі даних. Для того щоб користувач міг надати облікові дані, у системі існує сторінка входу з набором полів, з якими користувач може взаємодіяти.

JWT являє собою веб-стандарт, який визначає спосіб передачі даних про користувача в форматі JSON в зашифрованому вигляді.

Сервер автентифікації перевіряє дані користувачів та генерує для них JWT. Таким чином, веб-система буде розпізнавати конкретного користувача за допомогою отриманого JWT. Схема роботи JWT автентифікації зображена на рисунку 4.5.

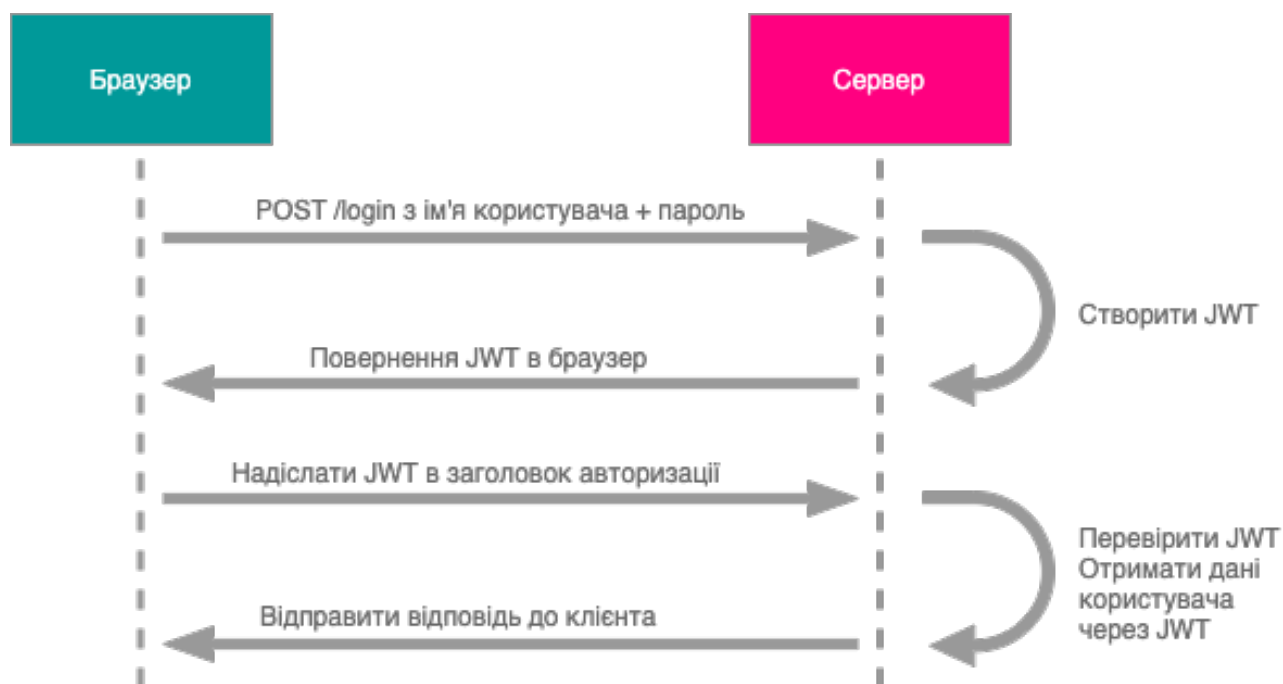


Рисунок 4.5 — Схема роботи JWT автентифікації

JWT не використовують сесии і не мають проблем з архітектурою мікрослужб. Замість того щоб створити сесии і встановити файл cookie, сервер відправить веб-маркер JSON. Далі ми можемо використовувати цей маркер, щоб робити все, що ми хочемо зробити з сервером.

Основні переваги JWT:

- Відсутність сесии для управління (без стану): JWT — це автономний маркер, який містить інформацію автентифікації, інформацію про час закінчення терміну дії та інші визначені користувачем твердження з цифровим підписом;

- Ніяких файлів cookie не потрібно, тому це дуже зручно для мобільних пристроїв;

- Хороша продуктивність: це скорочує час роботи мережі в обидва кінці;

- Маркер може бути згенерований в будь-якому місці. Автентифікація може відбуватися на сервері ресурсів або легко відділитися на свій власний сервер.

5. РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

Для роботи веб-системи повинен бути підключений інтернет та на комп'ютері користувача має бути встановлений будь-який браузер.

Відкриття системи відбувається за допомогою введення запиту у рядку адреси або завдяки переходу за посиланням: <https://diplom20200410104007.azurewebsites.net/map>.

Після запуску веб-системи користувач потрапить на головну сторінку системи обліку надзвичайних ситуацій соціально-політичного характеру (рисунок 5.1).

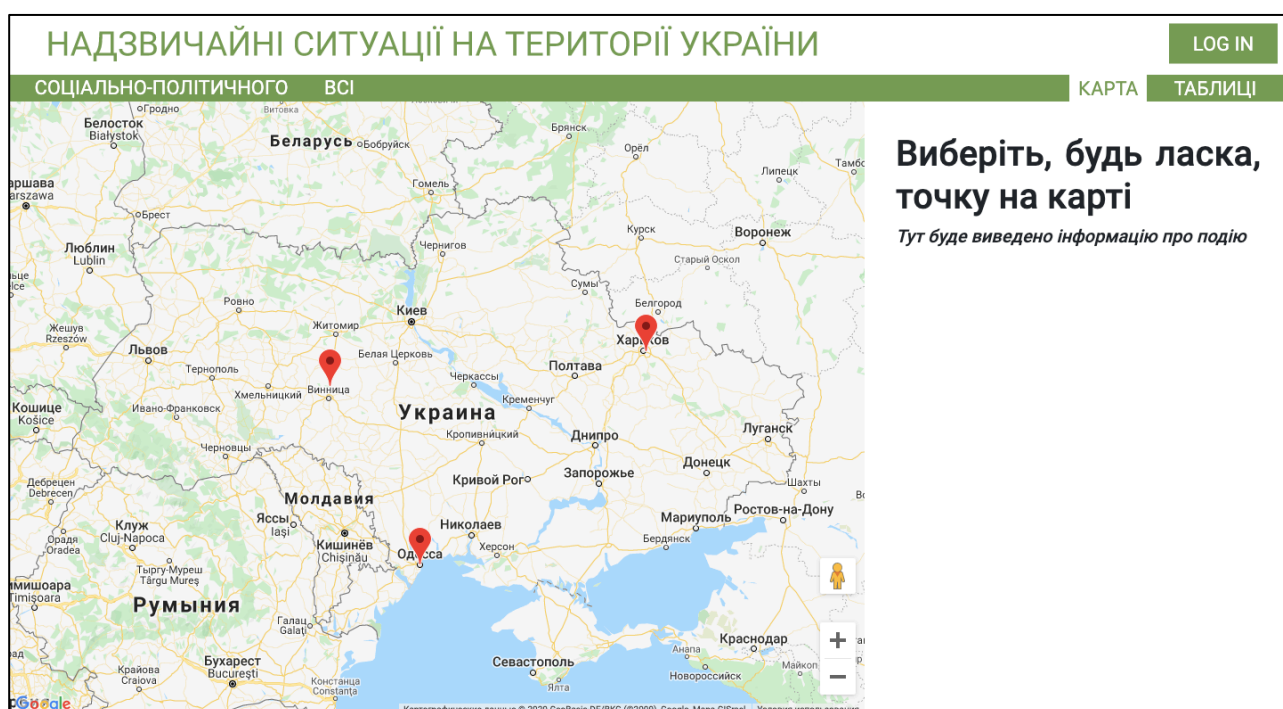


Рисунок 5.1 — Головна сторінка веб-системи

Щоб переглянути інформацію певної надзвичайної ситуації потрібно натиснути на червоний прапорець на карті, завдяки чому на екрані з'явиться опис вибраної надзвичайної ситуації (рисунок 5.2).

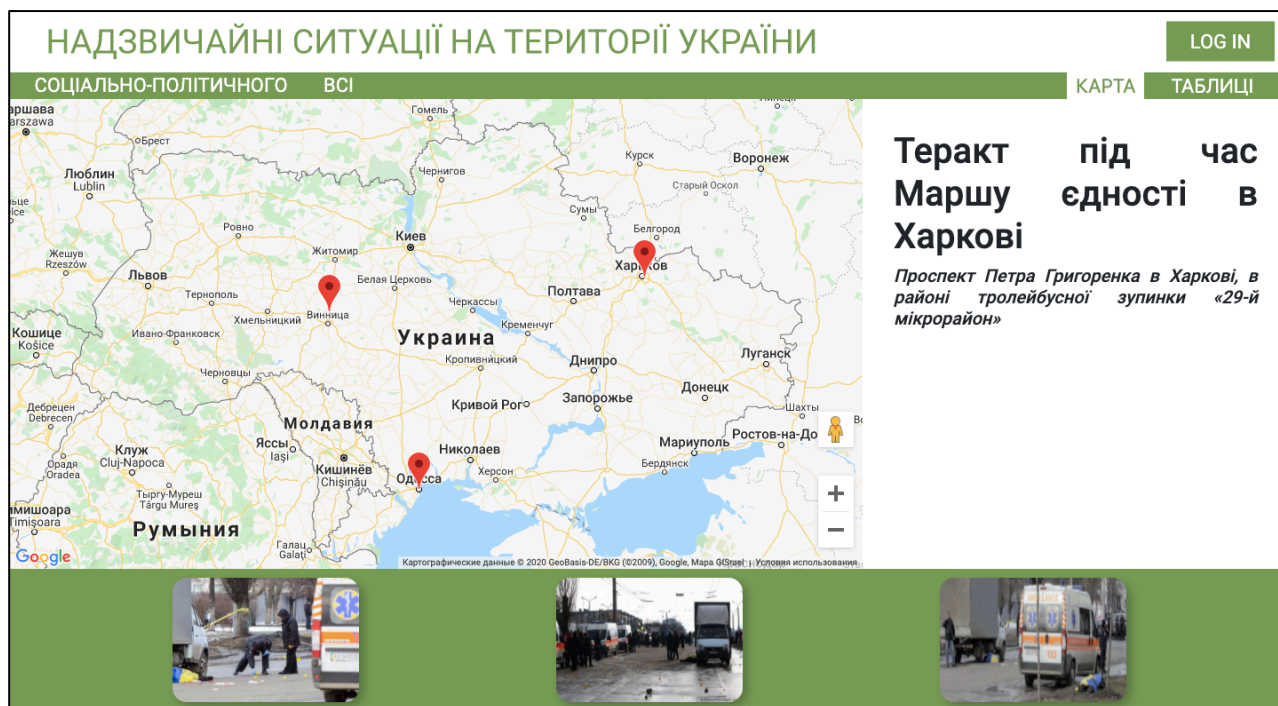


Рисунок 5.2 — Відображення певної НС соціально-політичного характеру

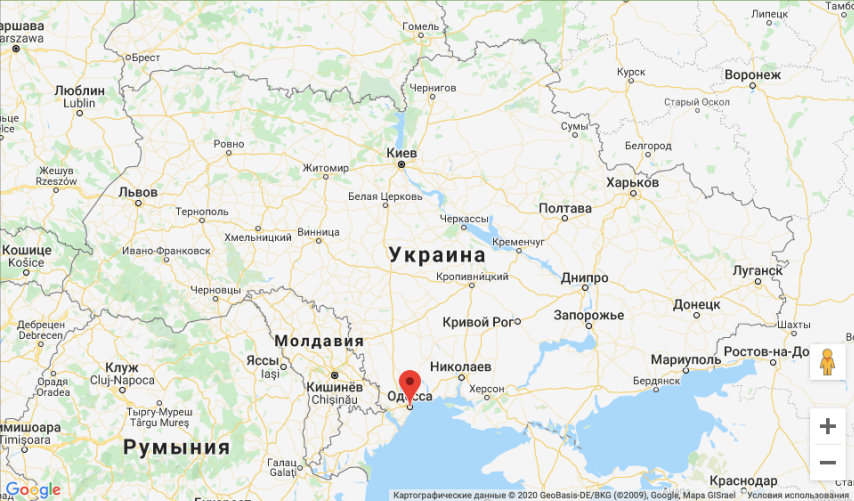
Над інтерактивною картою знаходиться панель керування зеленого кольору, з лівої сторони якої розміщені: назва характеру надзвичайних ситуацій та кнопка вибору року виникнення надзвичайних ситуацій. За замовчуванням параметр вибору року встановлений на показ усіх надзвичайних ситуацій за всі роки.

Для того щоб показати надзвичайні ситуації за певний рік, наприклад 2014, потрібно натиснути на параметр “ВСІ” та обрати “2014” (рисунок 5.3).

НАДЗВИЧАЙНІ СИТУАЦІЇ НА ТЕРИТОРІЇ УКРАЇНИ

LOG IN

СОЦІАЛЬНО-ПОЛІТИЧНОГО 2014 КАРТА ТАБЛИЦІ



Пожежа в будинку профсоюзів

Куликове поле

Після перемоги проукраїнських сил в центрі міста провокатори відійшли на південь у напрямку табору проросійських активістів на Куликовому полі, який знаходиться біля місцевого залізничного вокзалу. Колишні учасники проукраїнського маршу послідували за (як вони вважали) своїми нападниками, у результаті чого, дійшовши до майдану, де декілька місяців відбувалися проросійські мітинги, було спалено усі намети проросійських мітингарів. Після цього активісти з Куликового поля забарикадувалися у Будинку профспілок Одеської області, який знаходиться на тій же площі[24]. Через деякий час у будівлі, у яку увірвалися проросійські активісти, в результаті підпалу виникла пожежа, внаслідок чого загинуло кілька десятків осіб




Рисунок 5.3 — Відображення НС за 2014 рік

Окрім інтерактивної карти, користувач може переглянути довідкові дані щодо надзвичайних ситуацій у формі таблиць, натиснувши на панелі керування “ТАБЛИЦІ”. За замовчуванням інформація надається за усі роки (рисунок 5.4).

НАДЗВИЧАЙНІ СИТУАЦІЇ НА ТЕРИТОРІЇ УКРАЇНИ

LOG IN

КАРТА ТАБЛИЦІ

Довідкові дані щодо кількості постраждалих та загиблих осіб:

Всього НС	Характер НС	Загинуло, осіб	Постраждало, осіб
	соціального		
3	3	41	177

Надзвичайні ситуації: Характер НС: Рік:

Назва НС	Дата	Місце, де виникла НС	Збитки	Вартість ліквідації
Вибухи на складах під Вінницею	Jun 1, 2017	Склади під Калинівкою	0	100000000
Пожежа в будинку профсоюзів	Jan 2, 2014	Куликове поле	0	0
Теракт під час Маршу єдності в Харкові	Oct 5, 2017	Проспект Петра Григоренка в Харкові, в районі тролейбусної зупинки «29-й мікрорайон»	0	0

Рисунок 5.4 — Сторінка “ТАБЛИЦІ” за замовчуванням

Перша таблиця містить загальну кількість надзвичайних ситуацій у системі та кількість усіх постраждалих та загиблих осіб. У другій таблиці представлено назву НС, дату виникнення, місце виникнення, збитки та вартість ліквідації.

Користувачу надається змога переглянути надзвичайні ситуації за певний рік, натиснувши на поле “ВСІ” біля надпису “Рік” та обравши, наприклад, “2017” (рисунок 5.5).

НАДЗВИЧАЙНІ СИТУАЦІЇ НА ТЕРИТОРІЇ УКРАЇНИ
LOG IN

КАРТА
ТАБЛИЦІ

Довідкові дані щодо кількості постраждалих та загиблих осіб:

Всього НС	Характер НС	Загинуло, осіб	Постраждало, осіб
	соціальний		
3	3	41	177

Надзвичайні ситуації: Характер НС: Рік:

Назва НС	Дата	Місце, де виникла НС	Збитки	Вартість ліквідації
Вибухи на складах під Вінницею	Jun 1, 2017	Склади під Калинівкою	0	100000000
Теракт під час Маршу єдності в Харкові	Oct 5, 2017	Проспект Петра Григоренка в Харкові, в районі тролейбусної зупинки «29-й мікрорайон»	0	0

Рисунок 5.5 — Вибір НС за 2017 рік у таблиці

Також система має форму підрахунку суми збитків надзвичайної ситуації від втрати життя та здоров'я населення під час терористичних актів, таких як масові вбивства (рисунок 5.6).

НАДЗВИЧАЙНІ СИТУАЦІЇ НА ТЕРИТОРІЇ УКРАЇНИ LOG IN

Розрахунок збитків від втрати життя та здоров'я населення

Кількість працівників з втратою працездатності до 9 днів

Кількість працівників з втратою працездатності понад 9 днів

Кількість працівників з отриманою інвалідністю та втратою працездатності понад 11 років

Загальна кількість загиблих

Кількість загиблих до 16 років

Кількість загиблих від 16 до 60 років

Середній вік дітей, що залишилися без годувальника

Кількість дітей, що залишилися без годувальника

75 тис. грн.

Рисунок 5.6 — Форма підрахунку суми збитків НС від втрати життя та здоров'я населення

У системі також передбачений вхід для адміністратора, який має доступ до внесення нових та редагування існуючих надзвичайних ситуацій. Щоб увійти у систему потрібно натиснути кнопку “LOG IN” у верхньому кутку з правої сторони. Одразу відкриється вікно авторизації (рисунок 5.7).

НАДЗВИЧАЙНІ СИТУАЦІЇ НА ТЕРИТОРІЇ УКРАЇНИ

LOG IN

КАРТА ТАБЛИЦІ

Login

Email

Email

Password

Password

Login

Рисунок 5.7 — Вікно авторизації

Система перевіряє правильність введених даних для входу.

У разі введення електронної пошти в некоректному форматі, на екрані з'являється відповідне повідомлення (рисунок 5.8).

Login

Email

admin

Please enter a valid email

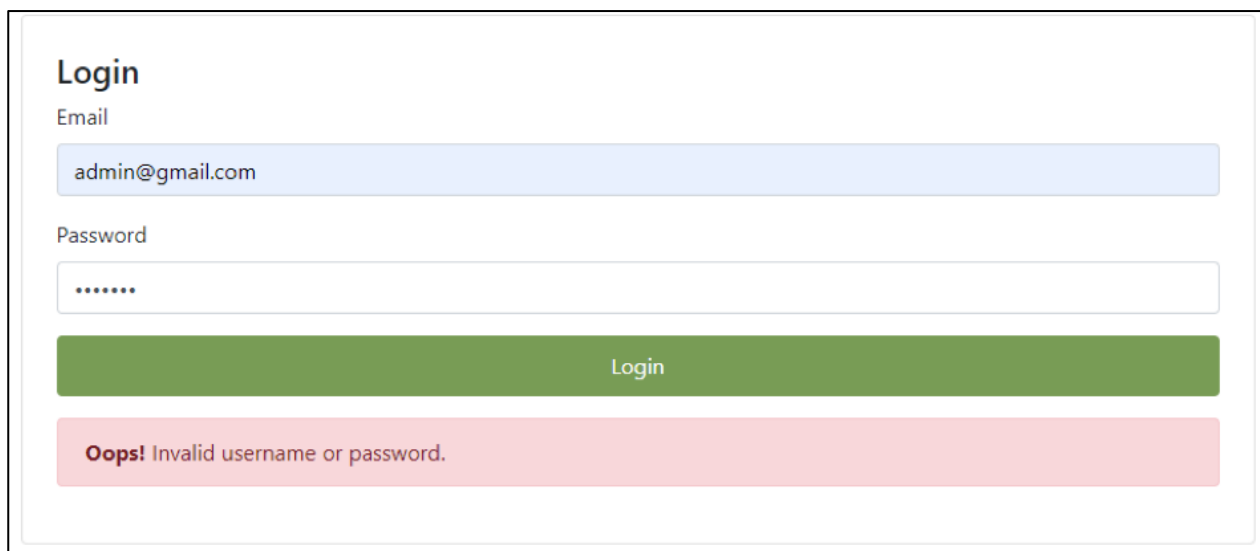
Password

Password

Login

Рисунок 5.8 — Повідомлення про некоректно введenu електронну пошту

Якщо введена неправильна електронна пошта або пароль, на екрані з'являється відповідне повідомлення (рисунок 5.9).

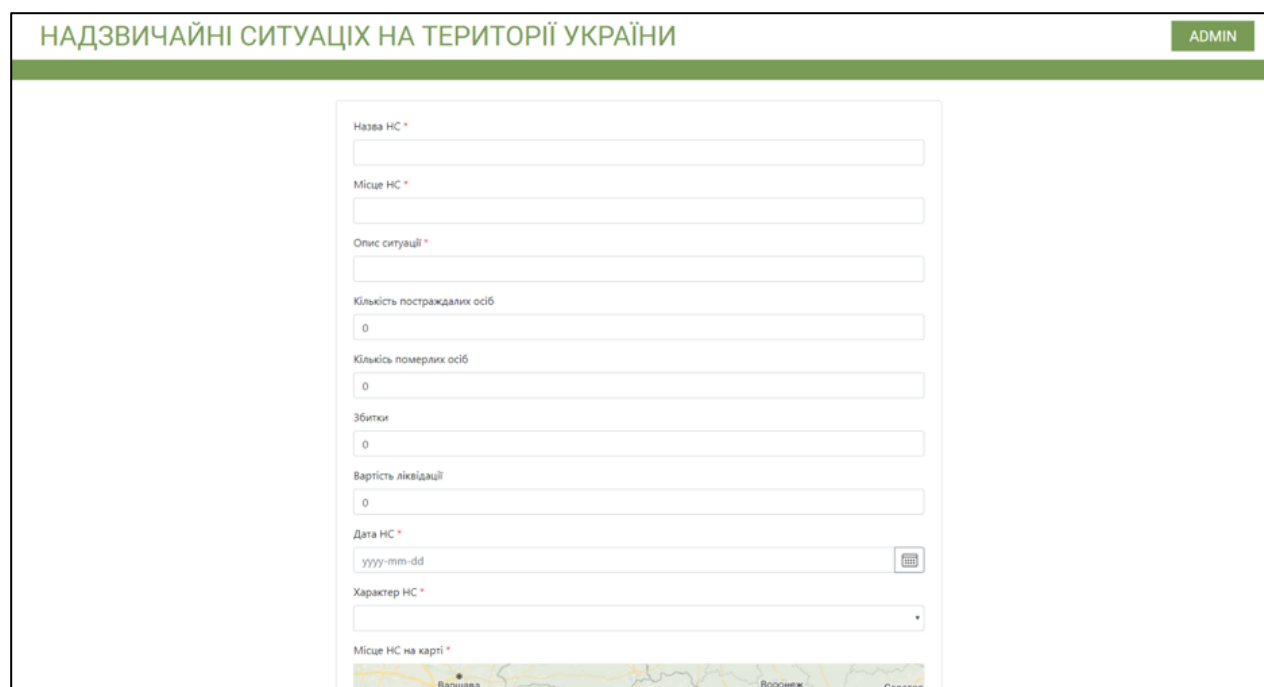


The screenshot shows a login interface with the following elements:

- Header:** "Login" in bold black text.
- Email:** A text input field containing "admin@gmail.com".
- Password:** A text input field with masked characters "*****".
- Login Button:** A green button with the text "Login".
- Error Message:** A pink banner below the button containing the text "Oops! Invalid username or password."

Рисунок 5.9 — Повідомлення про неправильну електронну пошту або пароль

Внаслідок успішного входу до системи як адміністратора відкривається сторінка додання нової надзвичайної ситуації (рисунок 5.10).



The screenshot shows a web form titled "НАДЗВИЧАЙНІ СИТУАЦІЇ НА ТЕРИТОРІЇ УКРАЇНИ" with an "ADMIN" button in the top right corner. The form contains the following fields:

- Назва НС ***: Text input field.
- Місце НС ***: Text input field.
- Опис ситуації ***: Text input field.
- Кількість постраждалих осіб**: Text input field with "0" entered.
- Кількість померлих осіб**: Text input field with "0" entered.
- Збитки**: Text input field with "0" entered.
- Вартість ліквідації**: Text input field with "0" entered.
- Дата НС ***: Date picker field showing "yyyy-mm-dd".
- Характер НС ***: Dropdown menu.
- Місце НС на карті ***: Map area showing locations like "Київ", "Воронеж", and "Саратов".

Рисунок 5.10 — Сторінка додання нової надзвичайної ситуації

Також для додання нової НС існує кнопка "Додати НС" на сторінці "ТАБЛИЦІ" (рисунок 5.11).

НАДЗВИЧАЙНІ СИТУАЦІЇ НА ТЕРИТОРІЇ УКРАЇНИ

LOG OUT

КАРТА ТАБЛИЦІ

Довідкові дані щодо кількості постраждалих та загиблих осіб:

Всього НС	Характер НС	Загинуло, осіб	Постраждало, осіб
	соціальний		
3	3	41	177

Надзвичайні ситуації: Характер НС: Рік:

Назва НС	Дата	Місце, де виникла НС	Збитки	Вартість ліквідації
Вибухи на складах під Вінницею	Jun 1, 2017	Склади під Калинвікою	0	100000000
Пожежа в будинку профсоюзів	Jan 2, 2014	Куликове поле	0	0
Теракт під час Маршу єдності в Харкові	Oct 5, 2017	Проспект Петра Григоренка в Харкові, в районі тролейбусної зупинки «29-й мікрорайон»	0	0

Додати НС

Рисунок 5.11 — Сторінка “ТАБЛИЦІ” з кнопкою додання нової надзвичайної ситуації

Щоб додати дані про нову надзвичайну ситуацію обов’язково потрібно заповнити усі поля, інакше система виведе відповідні повідомлення (рисунок 5.12).

Назва НС *

Name is required

Місце НС *

Address is required

Опис ситуації *

Description is required

Рисунок 5.12 — Повідомлення про незаповнені поля

Для того, щоб зазначити місце, в якому сталась надзвичайна ситуація, потрібно

натиснути кнопкою миші на відповідний регіон на карті (рисунок 5.13).

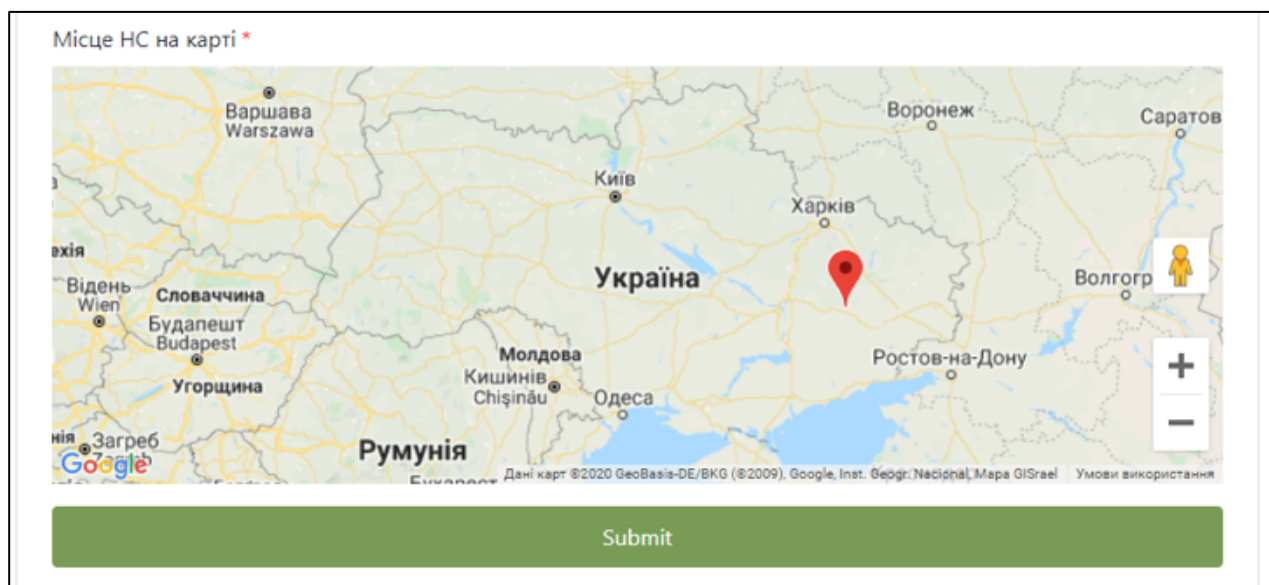


Рисунок 5.13 — Вибір місця НС на карті

Переконавшись, що всі поля заповнені правильно, потрібно натиснути на кнопку “Submit” для підтвердження створення.

Результат додання нової надзвичайної ситуації буде представлений на карті (рисунок 5.14) і в таблицях.

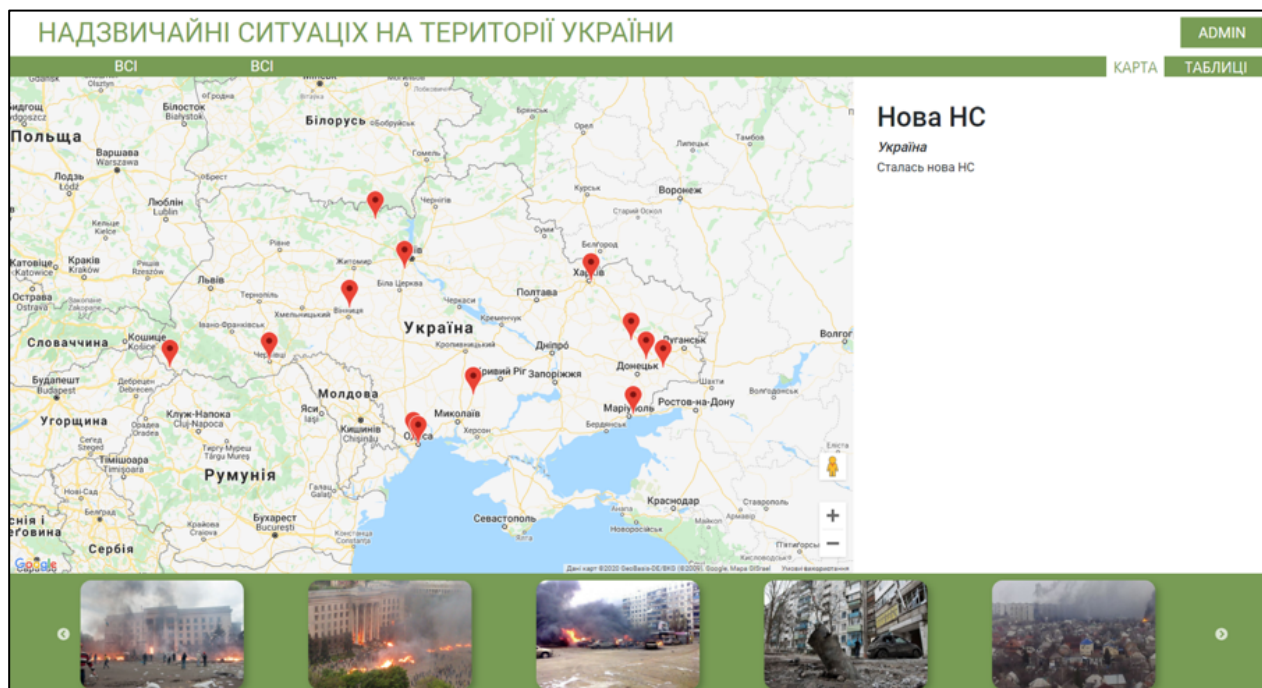


Рисунок 5.14 — Результат додання НС на карті

Для випадків, коли під час додання нової надзвичайної ситуації, були введені помилкові дані або з'явилися нові, передбачено можливість редагування надзвичайних ситуацій в системі. На сторінках “КАРТА” та “ТАБЛИЦІ” адміністратора розташована кнопка редагування після інформації про обрану надзвичайну ситуацію (рисунок 5.15).



Рисунок 5.15 — Кнопка редагування

Після натиску на кнопку відкривається сторінка редагування НС (рисунок 5.16).

Скріншот веб-сторінки для редагування надзвичайної ситуації. У верхній частині заголовок "НАДЗВИЧАЙНІ СИТУАЦІЇ НА ТЕРИТОРІЇ УКРАЇНИ" та кнопки "LOG OUT", "КАРТА" та "ТАБЛИЦІ". Під заголовком навігаційна панель з "СОЦІАЛЬНО-ПОЛІТИЧНОГО" та "ВСІ". Основна частина сторінки містить форму з наступними полями: "Назва НС" (Теракт під час Маршу єдності в Харкові), "Місце НС" (Проспект Петра Григоренка в Харкові, в районі тролейбусної зупинки «29-й мік»), "Опис ситуації", "Кількість постраждалих осіб" (9), "Кількість померлих осіб" (4), "Збитки" (0), "Вартість ліквідації" (0) та "Дата НС".

Назва НС *
Теракт під час Маршу єдності в Харкові
Місце НС *
Проспект Петра Григоренка в Харкові, в районі тролейбусної зупинки «29-й мік»
Опис ситуації *
Кількість постраждалих осіб
9
Кількість померлих осіб
4
Збитки
0
Вартість ліквідації
0
Дата НС *

Рисунок 5.16 — Сторінка редагування НС

Отже, система не потребує наявності спеціального програмного забезпечення, є простою та не вимагає спеціальних технічних знань. Систему можуть використовувати як звичайні користувачі, так і адміністратори.

ВИСНОВКИ

В ході виконання роботи було розроблено веб-систему обліку надзвичайних ситуацій соціально-політичного характеру.

Було проведено аналіз проблеми розробки інтерактивної карти надзвичайних ситуацій та були досліджені існуючі веб-системи даного напрямку.

Було проведено аналіз програмних засобів і методів, використаних під час розробки системи.

Розроблена система дозволяє користувачам взаємодіяти з інтерактивною картою. Користувачами системи можуть бути звичайні користувачі та адміністратори, які відповідають за додання та правильність інформації про надзвичайні ситуації. Система може працювати на будь-якій операційній системі, де є браузер та доступ до інтернету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Класифікатор надзвичайних ситуацій ДК 019:2010 [Електронний ресурс] // Держспоживстандарт України. — 2010. — Режим доступу: <https://zakon.rada.gov.ua/rada/show/va457609-10>.

2. Онищенко М. Г. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ КАРТОГРАФІЧНОГО МОДЕЛЮВАННЯ ПРИРОДНО-ТЕХНОГЕННОЇ БЕЗПЕКИ УКРАЇНИ [Електронний ресурс] / М. Г. Онищенко // Державне науково-виробниче підприємство «Картографія». — 2017. — Режим доступу: http://maptimes.inf.ua/CH_18/Ch18_Article1_Natural-antropogenic-safety.html.

3. Social networks perspective of firefighters' adaptive behaviour and coordination among them [Електронний ресурс] / A. Abbasi, L. Hossain, J. Hamra, C. Owen // ACM International Conference on Cyber, Physical and Social Computing. — 2010. — Режим доступу: <https://dl.acm.org/doi/pdf/10.1109/GreenCom-CPSCoM.2010.57>.

4. Geo-twitter analytics: Applications in crisis management / [A. MacEachren, A. Robinson, A. Jaiswal та ін.]. — Paris, 2011. — (25th international cartographic conference proceedings).

5. Alexander D. An interpretation of disaster in terms of changes in culture, society and international relations / D. Alexander, R. W. Perry, E. Quarantelli // What is a disaster? New answers to old questions / D. Alexander, R. W. Perry, E. Quarantelli., 2005.

6. Barton A. H. Disaster and collective stress / A. H. Barton, R. W. Perry, E. Quarantelli // What is a disaster? New answers to old questions / A. H. Barton, R. W. Perry, E. Quarantelli., 2005. — С. 125–152.

7. Palen L. Citizen communications in crisis: Anticipating a future of ICT-supported public participation [Електронний ресурс] / L. Palen, S. B. Liu // Proceedings of the SIGCHI conference on human factors in computing systems. — 2007. — Режим доступу: <http://www.hcitang.org/uploads/Teaching/palenliu-chi07.pdf>.

8. Clarke K. C. On Epidemiology and Geographic Information Systems: A Review and Discussion of Future Directions [Электронный ресурс] / К. С. Clarke, S. L. McLafferty, J. T. Barbara. — 1996. — Режим доступа: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2639830/pdf/8903207.pdf>.

9. Environmental Risk Factors for Lyme Disease Identified with Geographic Information Systems / [G. E. Glass, B. S. Schwartz, J. M. Morgan та ін.]. // American Journal of Public Health. — 1995. — №85. — С. 944–948.

10. Remote Sensing as a Landscape Epidemiologic Tool to Identify Villages at High Risk for Malaria Transmission / [L. R. Beck, M. H. Rodrigues, S. W. Dister та ін.]. // The American Journal of Tropical Medicine and Hygiene. — 1994. — №51. — С. 80–271.

11. Global Monitoring and Disease Prediction Program / B. L. Wood, L. R. Beck, S. W. Dister, M. A. Spanner. // Sistema Terra. — 1994. — №3. — С. 42.

12. Chakraborty J. Using Geographic Plume Analysis to Assess City of Winston–Salem Integrated Network Fire Operations Project / J. Chakraborty, M. P. Armstrong., 1996.

13. Improving Disaster Management: The Role of IT in Mitigation, Preparedness, Response, and Recovery / [T. Schmitt, J. Eisenberg, R. Rao та ін.]. — Washington, D.C.: National Academies Press, 2007.

14. Volunteer technology communities: open development. The World Bank and GFDRR — Washington, D.C.: GFDRR Labs, 2011.

15. Moore R. Hashtag standards for emergencies [Электронный ресурс] / R. Moore, A. Verity // Office for the Coordination of Humanitarian Affairs. — 2014. — Режим доступа: <https://www.unocha.org/sites/unocha/files/Hashtag%20Standards%20for%20Emergencies.pdf>.

16. Sancho G. R. Networks, insurgencies, and prefigurative politics: A cycle of global indignation [Электронный ресурс] / Sancho // Convergence. — 2014. — Режим доступа: https://www.researchgate.net/publication/275484689_Networks_insurgencies_and_prefigurative_politics_A_cycle_of_global_indignation.

17. Stadler F. Between democracy and spectacle: the front-end and back-end of the social web / Stadler // The Social Media Reader / Stadler. — New York and London: New York University Press, 2012.

18. Stiegler R. Finding family and friends in the aftermath of a disaster using federated queries on social networks and websites / R. Stiegler, S. Tilley, T. Parveen. — Williamsburg: IEEE, 2011.

19. Skinner J. Natural disasters and twitter: Thinking from both sides of the tweet [Електронний ресурс] / Skinner. — 2013. — Режим доступу: <https://journals.uic.edu/ojs/index.php/fm/article/view/4650/3741>.

20. Communicating with the public about emerging health threats: Lessons from the pre-event message development project / [R. J. Wray, S. M. Becker, N. Henderson та ін.]. // American Journal of Public Health. — 2008. — С. 2214–2222.

21. Procopio C. H. Do You Know What It Means to Miss New Orleans? Internet Communication, Geographic Community, and Social Capital in Crisis [Електронний ресурс] / C. H. Procopio, S. T. Procopio. — 2007. — Режим доступу: <https://doi.org/10.1080/00909880601065722>.

22. Retrieving Information from Multiple Sources [Електронний ресурс] / [A. Roy, K. Ghosh, M. Basu та ін.] // The 2018 Web Conference Companion. — 2018. — Режим доступу: <https://dl.acm.org/doi/fullHtml/10.1145/3184558.3186920>.

23. Analysis and early detection of rumors in a post disaster scenario [Електронний ресурс] / [T. Mondal, P. Pramanik, I. Bhattacharya та ін.] // Information Systems Frontiers. — 2018. — Режим доступу: <https://doi.org/10.1007/s10796-018-9837-8>.

24. Characterizing and Countering Communal Microblogs during Disaster Events / K. Rudra, A. Sharma, N. Ganguly, S. Ghosh. // IEEE Transactions on Computational Social Systems. — 2018. — С. 403–417.

25. National Research Council. Successful Response Starts with a Map: Improving Geospatial Support for Disaster Management [Електронний ресурс] / National Research Council // The National Academies Press. — 2007. — Режим доступу: <https://doi.org/10.17226/11793>.

26. Гнатюк С. Досвід і перспективи використання технологій краудсорсингу у надзвичайних ситуаціях. Аналітична записка [Електронний ресурс] / С. Гнатюк — Режим доступу: <https://niss.gov.ua/doslidzhennya/nacionalna-bezpeka/dosvid-i-perspektivi-vikoristannya-tekhnologiy-kraudsorsingu-u>.

27. История стартапа: как интерактивная карта событий в Украине Liveuamap превратилась в мировой медиаресурс [Электронный ресурс]. — Режим доступа: <http://ain.ua/2015/01/18/558878>.
28. RSOE — Emergency and Disaster Information Service [Электронный ресурс]. — Режим доступа: <http://hisz.rsoe.hu/>.
29. Copernicus Emergency Management Service — Mapping [Электронный ресурс]. — Режим доступа: <https://emergency.copernicus.eu/mapping/>.
30. PDC DisasterAWARE [Электронный ресурс]. — Режим доступа: <https://disasteralert.pdc.org/disasteralert/>.
31. Sole A. Visual Studio Code Distilled: Evolved Code Editing for Windows, macOS, and Linux / Alessandro Sole., 2019.
32. Posadas M. Mastering C# and .NET Framework / Marino Posadas., 2016.
33. Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. / Джеффри Рихтер., 2013.
34. Vemula R. Real-Time Web Application Development: With ASP.NET Core, SignalR, Docker, and Azure / Rami Vemula., 2017.
35. Lobel L. G. Microsoft Azure SQL Database Step by Step / L. G. Lobel, E. D. Boyd., 2014.
36. Rahmatulloh A. Performance comparison of signed algorithms on JSON Web Token [Электронный ресурс] / Rahmatulloh. — 2019. — Режим доступа: <https://iopscience.iop.org/article/10.1088/1757-899X/550/1/012023/pdf>.
37. Williamson K. Learning AngularJS: A Guide to AngularJS Development / Ken Williamson., 2015.
38. Konarski M. Using Google Maps API along with technology .NET [Электронный ресурс] / M. Konarski, W. Zabierowski. — 2010. — Режим доступа: https://www.researchgate.net/publication/251923396_Using_Google_Maps_API_along_with_technology_NET.

ДОДАТОК А

Система обліку надзвичайних подій соціально-політичного характеру.

Специфікація

УКР.НТУУ"КПІ ім. Ігоря Сікорського"_ТЕФ_АПЕПС_ТМ62200_20Б

Аркушів 1

Київ 2020

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ”КПІ ім. Ігоря Сікорського”_ТЕФ_АПЕПС_ ТМ62200_20Б	Записка.docx	Пояснюв альна записка
Компоненти		
УКР.НТУУ”КПІ ім. Ігоря Сікорського”_ТЕФ_АПЕПС_ ТМ62200_20Б 12-1	EventsController .cs main- map.component.js add- event.component.js calc- loss.component.js	Основні компоненти
УКР.НТУУ”КПІ ім. Ігоря Сікорського”_ТЕФ_АПЕПС_ ТМ62200_20Б 13-1	Додаток В.docx	Опис програмного модуля

ДОДАТОК Б

Система обліку надзвичайних подій соціально-політичного характеру.

Текст програми

УКР.НТУУ"КПІ ім. Ігоря Сікорського"_ТЕФ_АПЕПС_ТМ62200_20Б 12-1

Аркушів 10

Київ 2020


```

using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc;
using System.Threading.Tasks;
using Diplom.DataModels;
using Microsoft.AspNetCore.Http;
using Diplom.Services;
using System;
using Microsoft.AspNetCore.Authorization;
using Diplom.ViewModels;
using System.Text.RegularExpressions;
using Newtonsoft.Json.Linq;
using Newtonsoft.Json;

namespace Diplom.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class EventsController : ControllerBase
    {
        private readonly IEventService _eventService;

        public EventsController(IEventService eventService)
        {
            _eventService = eventService;
        }

        [HttpGet]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<IEnumerable<Event>> GetAsync()
        {
            return await _eventService.ListAsync();
        }

        [HttpGet("{eventId}")]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<IEnumerable<Event>> GetAsync(int eventId)
        {
            return await _eventService.ListAsync(eventId);
        }

        [HttpGet("dates/{date}")]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<IEnumerable<Event>> GetAsync(DateTime date)
        {

```

```

    return await _eventService.ListAsync(date);
}

[HttpGet("{date}/{emergencyId}")]
[ProducesResponseType(StatusCodes.Status200OK)]
public async Task<IEnumerable<Event>> GetAsync(DateTime date, int
emergencyId)
{
    return await _eventService.ListAsync(date, emergencyId);
}

[HttpGet("dates")]
[ProducesResponseType(StatusCodes.Status200OK)]
public async Task<List<DateTime?>> GetDatesAsync()
{
    return await _eventService.DatesListAsync();
}

[HttpPut("update")]
[ProducesResponseType(StatusCodes.Status200OK)]
public async Task<ActionResult> UpdateEventAsync([FromBody] JObject
jEventDTO)
{
    JsonSerializer serializer = new JsonSerializer();
    EventDTO eventDTO = (EventDTO)serializer.Deserialize(new
JTokenReader(jEventDTO), typeof(EventDTO));
    try
    {
        var result = await _eventService.UpdateEventAsync(eventDTO);

        if (!result.Success)
            return BadRequest(result.Message);
    }
    catch (Exception e)
    {
        return BadRequest(e.Message);
    }

    return Ok();
}

// POST: api/Event/add
[HttpPost("add"), DisableRequestSizeLimit]

```

```

[ProducesResponseType(StatusCodes.Status200OK)]
public async Task<ActionResult> AddEventAsync([FromBody] JObject jEventDTO)
{
    JsonSerializer serializer = new JsonSerializer();
    EventDTO eventDTO = (EventDTO)serializer.Deserialize(new
JTokenReader(jEventDTO), typeof(EventDTO));
    try
    {
        var result = await _eventService.AddEventAsync(eventDTO);

        if (!result.Success)
            return BadRequest(result.Message);

    }
    catch (Exception e)
    {
        return BadRequest(e.Message);
    }

    return Ok();
}

// POST: api/Event/delete
[HttpPost("delete"), ActionName("Delete")]
public async Task<IAActionResult> Delete(int eventId)
{
    var result = await _eventService.DeleteAsync(eventId);

    if (!result.Success)
        return BadRequest(result.Message);

    return Ok();
}
}
}

import { Component, OnInit, Input, OnDestroy } from '@angular/core';
import { Marker } from '../shared/models/Marker';
import { EventService } from '../shared/services/event.service';
import { Event } from '../shared/models/event';
import { Observable, Subscription, of } from 'rxjs';
import { UserService } from '../shared/services/user.service';
import { Router } from '@angular/router';

```

```

@Component({
  selector: 'app-main-map',
  templateUrl: './main-map.component.html',
  styleUrls: ['./main-map.component.scss']
})
export class MainMapComponent implements OnInit, OnDestroy {
  constructor(private eventService: EventService,
    private userService: UserService,
    private router: Router) { }
  status: boolean;
  subscription: Subscription;
  latitude: number;
  longitude: number;
  zoom: number;
  markers: Marker[];
  currentMarker: Marker;
  slides: { img: string }[][] = [];
  events: Event[]
  events$: Observable<Event[]>;
  @Input()
  ngOnInit() {
    this.markers = [];
    this.setInit();
    this.subscription = this.userService.authNavStatus$.subscribe(status => this.status =
status);
  }
  ngOnDestroy() {
    this.subscription.unsubscribe();
  }

  // Get Current Location Coordinates
  private setInit() {
    this.latitude = 49.045639;
    this.longitude = 31.159608;
    this.zoom = 6;
    this.setInitMarkers();
  }

  updateEvent(){
    let currentEvent = this.events.find(event => event.eventId ==
this.currentMarker.eventId);
    this.router.navigate(['/add-event'], {state: {existedEvent: currentEvent}});
  }
}

```

```

private setInitMarkers() {
  this.eventService.events.subscribe(events => {
    this.events = events;
    this.markers = events.filter(event => event.eventPosition).map((event) => {
      let slides = [];
      if(event.imageData){
        const images = event.imageData.split("|");
        console.log(images);
        images.forEach(img => slides.push({img: img}))
      }
      return <Marker>{ lat: event.eventPosition.x, lng: event.eventPosition.y, name:
event.eventName, address: event.eventPosition.place, desc: event.description, eventId:
event.eventId, slides: slides } });
      console.log(this.markers, this.events)
    });
  }

selectMarker(marker: Marker) {
  this.currentMarker = marker;
  console.log(this.currentMarker)
}

slideConfig = {
  slidesToScroll: 3,
  slidesToShow: 4,
  responsive: [
    {breakpoint: 1220, settings: {slidesToScroll: 2, slidesToShow: 3}},
    {breakpoint: 980, settings: {slidesToScroll: 2, slidesToShow: 3}},
    {breakpoint: 740, settings: {slidesToScroll: 1, slidesToShow: 2}},
    {breakpoint: 500, settings: {slidesToScroll: 1, slidesToShow: 1}}
  ]
};

slickInit(e) {
  console.log('slick initialized');
}

breakpoint(e) {
  console.log('breakpoint');
}

afterChange(e) {
  console.log('afterChange');
}

beforeChange(e) {
  console.log('beforeChange');
}

```

```

}

import { Component, OnInit, Input } from '@angular/core';
import { Marker } from '../shared/models/Marker';
import { Event } from '../shared/models/event';
import { NgbDateParserFormatter, NgbDate } from '@ng-bootstrap/ng-bootstrap';
import { EMERGENCY_FE_NAME_TO_ID, EMERGENCY_FE_ID_TA_NAME }
from '../shared/models/emergency';
import { EventService } from '../shared/services/event.service';
import { ActivatedRoute, Router } from '@angular/router';
import { map } from 'rxjs/operators'
import { HttpClient } from '@angular/common/http';

@Component({
  selector: 'app-add-event',
  templateUrl: './add-event.component.html',
  styleUrls: ['./add-event.component.scss']
})
export class AddEventComponent implements OnInit {
  constructor( private ngbDateParserFormatter: NgbDateParserFormatter,
    private eventService: EventService,
    private activatedRoute: ActivatedRoute,
    private router: Router,
    private http: HttpClient) { }
  isExistedEvent: boolean = false;

  newMarker: Marker;
  currentEvent: Event;
  latitude: number;
  longitude: number;
  zoom: number;
  currentEventDate: NgbDate;
  emergencyType: string;
  selectedFiles: {name: string, data: string}[] = []

  ngOnInit() {
    this.newMarker = new Marker();
    this.newMarker = { lat: 0, lng: 0,name: "", address: "", desc: "", eventId: null };
    this.currentEvent = new Event();

    // this.currentEventDate = new Date('2020');
    // console.log(this.currentEventDate.toISOString(), this.currentEventDate)
    this.latitude = 49.045639;
    this.longitude = 31.159608;
  }

```

```

this.zoom = 5.2;
this.activatedRoute.paramMap
  .pipe(map(() => window.history.state))
  if(window.history.state.existedEvent){
    this.setCurrentEvent(window.history.state.existedEvent);
    this.isExistedEvent = true
  }
  this.currentEvent.imageByteArrayList = this.currentEvent.imageByteArrayList || [];
}

setCurrentEvent(event: Event){
  this.currentEvent = event;
  let date = new Date(event.date)
  this.currentEventDate = new NgbDate(date.getFullYear(), date.getMonth(),
date.getDay());
  this.newMarker = <Marker>{ lat: event.eventPosition.x, lng: event.eventPosition.y,
name: event.eventName, address: event.eventPosition.place, desc: event.description,
eventId: event.eventId }
  this.emergencyType =
EMERGENCY_FE_ID_TA_NAME.get(event.emergencyId).toLowerCase()
}

addMarker(lat: number, lng: number) {
  this.newMarker.lat = lat;
  this.newMarker.lng = lng;
  console.log(this.newMarker)
}

selectedFile: File

onFileChanged(event) {
  this.selectedFile = event.target.files[0];
  console.log(event)
  this.onUpload()
}

onUpload() {
  console.log(this.selectedFile)
  var reader = new FileReader();
  let thiss = this
  reader.onload = function () {
    console.log(reader.result);
    thiss.selectedFiles.push({name: thiss.selectedFile.name, data: <string> reader.result})
  }
}

```

```

    reader.readAsDataURL(this.selectedFile);
  }
  removeSelectedFile(file) {
    const index = this.selectedFiles.indexOf(file);
    if(index > -1){
      this.selectedFiles.splice(index, 1);
    }
  }
}

onSubmitted() {
  this.currentEvent.harmed = (this.currentEvent.harmed && this.currentEvent.harmed > 0
) ? this.currentEvent.harmed : 0;
  this.currentEvent.deaths = (this.currentEvent.deaths && this.currentEvent.deaths > 0 ) ?
this.currentEvent.deaths : 0;
  this.currentEvent.losses = (this.currentEvent.losses && this.currentEvent.losses > 0 ) ?
this.currentEvent.losses : 0;
  this.currentEvent.costs = (this.currentEvent.costs && this.currentEvent.costs > 0 ) ?
this.currentEvent.costs : 0;
  this.currentEvent.date = new
Date(this.ngbDateParserFormatter.format(this.currentEventDate)).toISOString();
  this.currentEvent.emergencyId =
EMERGENCY_FE_NAME_TO_ID.get(this.emergencyType.toUpperCase());
  this.currentEvent.eventPosition.x = this.newMarker.lat
  this.currentEvent.eventPosition.y = this.newMarker.lng
  console.log(this.newMarker, this.currentEvent);
  this.currentEvent.imageByteArrayList = this.selectedFiles.map(file => file.data)
  console.log(this.currentEvent.imageByteArrayList)
  this.currentEvent.emergency = null;

  this.eventService.modifyEvent(this.currentEvent, this.isExistedEvent);
  // this.router.navigate(["/map"])
}
}

import { Component, OnInit, Input } from '@angular/core';
import { Marker } from '../shared/models/Marker';
import { Event } from '../shared/models/event';
import { NgbDateParserFormatter, NgbDate } from '@ng-bootstrap/ng-bootstrap';
import { EMERGENCY_FE_NAME_TO_ID, EMERGENCY_FE_ID_TA_NAME }
from '../shared/models/emergency';
import { EventService } from '../shared/services/event.service';
import { ActivatedRoute, Router } from '@angular/router';
import { map } from 'rxjs/operators'
import { HttpClient } from '@angular/common/http';

```



```

@Component({
  selector: 'app-calc-loss',
  templateUrl: './calc-loss.component.html',
  styleUrls: ['./calc-loss.component.scss']
})
export class CalcLossComponent implements OnInit {
  lifeAndHealth: {
    Nml: number;
    Nmt: number;
    Nmi: number;
    Nmz60: number;
    Nmz16: number;
    Nmdp: number;
    AvgVD: number;
    Nmvtg: number;
  } = {
    Nml: null,
    Nmt: null,
    Nmi: null,
    Nmz60: null,
    Nmz16: null,
    Nmdp: null,
    AvgVD: null,
    Nmvtg: null
  }

  lifeAndHealthRes: {
    Hr: number;
    Vtrr: number;
    Vdp: number;
    Vvtg: number;
  } = {
    Hr: 0,
    Vtrr: null,
    Vdp: null,
    Vvtg: null
  }

  destruction: {
    Fv: number;
    Fg: number;
    Pr: number;
    Prs: number;
  }

```

```

    Sn: number;
    Mdg: number;
  } = {
    Fv: null,
    Fg: null,
    Pr: null,
    Prs: null,
    Sn: null,
    Mdg: null
  }
  destructionRes: number = 0;
  overallRes: number;

  constructor() {}
  ngOnInit() {
    // this.lifeAndHealth = {};
    // Object.keys(this.lifeAndHealth).forEach(v => this.lifeAndHealth[v] = 0);
  }
  onSubmitted() {
    this.lifeAndHealthRes.Vtrr = + (this.lifeAndHealth.Nml || 0) * 0.28 +
    (this.lifeAndHealth.Nmt||0) * 6.5 + (this.lifeAndHealth.Nmi||0) * 37 +
    (this.lifeAndHealth.Nmz16||0) * 22 + (this.lifeAndHealth.Nmz60||0) * 47;
    this.lifeAndHealthRes.Vdp = +(this.lifeAndHealth.Nmdp||0) * 0.15;
    this.lifeAndHealthRes.Vvtg = 12 * 0.037 * (18 - (this.lifeAndHealth.AvgVD||0)) *
    (this.lifeAndHealth.Nmvtg||0);
    this.lifeAndHealthRes.Hr = + (this.lifeAndHealthRes.Vtrr||0) +
    (this.lifeAndHealthRes.Vdp||0) + (this.lifeAndHealthRes.Vvtg||0);
    console.log(this.lifeAndHealth)
    this.destructionRes = 0;
    Object.keys(this.destruction).forEach(v => this.destructionRes += +(this.destruction[v]
    || 0));
    this.overallRes = +this.destructionRes + this.lifeAndHealthRes.Hr;
  }
}

```

ДОДАТОК В

Система обліку надзвичайних подій соціально-політичного характеру.

Опис програми

УКР.НТУУ"КПІ ім. Ігоря Сікорського"_ТЕФ_АПЕПС_ТМ62200_20Б 13-1

Аркушів 8

Київ 2020

АНОТАЦІЯ

Розділ містить опис частини, яка слугує для роботи з інтерактивною веб-картою, що є основною одиницею програмного продукту, та забезпечує поєднання усіх інших компонентів разом в одну веб-сторінку. Вказано принцип роботи з API, який слугує для отримання потрібних даних з серверу та бази даних. Модуль написано мовою програмування C#, з використанням EntityFramework, та Javascript.

ЗМІСТ

1. ЗАГАЛЬНІ ВІДОМОСТІ	78
2. ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ	79
3. ОПИС ЛОГІЧНОЇ СТРУКТУРИ	80
4. ТЕХНІЧНІ ЗАСОБИ, ЩО ВИКОРИСТОВУЮТЬСЯ	81
5. ВИКЛИК І ЗАВАНТАЖЕННЯ	82
6. ВХІДНІ ТА ВИХІДНІ ДАНІ	83

ЗАГАЛЬНІ ВІДОМОСТІ

У додатку розглядається один з програмних модулів системи — модуль для роботи з веб-сторінкою з кодом УКР.НТУУ”КПІ ім. Ігоря Сікорського”_ТЕФ_АПЕПС_ТМ62200_20Б 12-1, що міститься у файлі EventsController.cs. Модуль реалізовано за допомогою С# та бібліотек VCL. Модуль призначений для управління системою, яка відповідають за керування даними (API).

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Призначенням модулю для роботи з веб-картою є відображення інформації з серверу та бази даних та безпосереднього прийняття даних введення та обмін інформацією із клієнтським інтерфейсом. Використання такого шаблону дозволяє створювати програмне забезпечення, де інтерфейс і логіка роботи модуля для бази даних та серверу є незалежними компонентами, що дає можливість використовувати його для зменшення навантаження на клієнтську частину системи.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Слідкувати за зміною інформації про підсистеми є головним завданням модуля. При запуску системи модуль повертає всю інформацію, яка міститься в базі даних, для відображення даних на користувацькому інтерфейсі. Також модуль оброблює інформацію, надаючи змогу вводити потрібні дані та проводити обчислення та відображати результати у вигляді таблиць та на карті.

ТЕХНІЧНІ ЗАСОБИ ЩО ВИКОРИСТОВУЮТЬСЯ

Модуль розроблено у середовищі розробки Microsoft Visual Studio Code, що забезпечує набір сервісних функцій та графічний діалог з користувачем, на комп'ютері, має встановлений браузер та підключення до інтернету. Для реалізації клієнтської частини було використано бібліотеки Angular. За допомогою елементів Angular клієнтська частина може з'єднуватись з сервером за допомогою API.

В якості провайдера самої інтерактивної карти було обрано відкритий сервіс GoogleMaps.

ВИКЛИК І ЗАВАНТАЖЕННЯ

Програмний модуль реалізований як окремий клас, який забезпечує існування клієнтської частини та бізнес-логіки окремо від одного, але разом із цим запуск обох компонентів відбувається одночасно.

Для використання цього модуля треба завантажити веб-сторінку, яка зробить потрібні запити та відобразить результат.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними для модуля є інформація, яку користувач вводить в додатку та місцеположення.

Вихідними даними програмного модуля є відображення збережених даних на інтерактивній веб-карті та у вигляді таблиць.