

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток аналізу збитків від техногенних та природніх катастроф»

за спеціальністю 122 «Комп'ютерні науки»,  
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-61/7 Шишкін Олексій Васильович

Кваліфікаційна робота бакалавра  
захищена на засіданні ЕК

з оцінкою \_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 р.

Науковий керівник

\_\_\_\_\_  
(підпис)

к.т.н., доц., Марченко А.В.  
(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

\_\_\_\_\_  
(підпис)

Шифрін Д. М.  
(науковий ступінь, вчене звання, прізвище та ініціали)

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

Студент \_\_\_\_\_  
(підпис)

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук  
Секція інформаційних технологій проектування  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**

Зав. секцією ІТП

\_\_\_\_\_ В. В. Шендрик  
«\_\_» \_\_\_\_\_ 2020 р.

## **З А В Д А Н Н Я**

### **1 НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

*Шшикін Олексій Васильович*

**1 Тема роботи** Web-додаток аналізу збитків від техногенних та природніх катастроф

**керівник роботи** Марченко Анна Вікторівна, к.т.н., доцент,

затверджені наказом по університету від «14» травня 2020 р. № 0576-III

**2 Строк подання студентом роботи** «1» червня 2020 р.

**3 Вхідні дані до роботи** Методика розрахунку наслідків лісової пожежі, методика оцінки обстановки під час землетрусу, методичні розробки для оцінювання еколого-економічного збитку від викиду НХР

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** 1) Аналіз предметної області, 2) Проектування web-додатку аналізу збитків від техногенних та природніх катастроф, 3) Розробка web-додатку аналізу збитків від техногенних та природніх катастроф

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** аналіз предметної області, мета і задачі дипломного проекту, Порівняння аналогів, функціональні вимоги до web-додатку, моделювання роботи web-додатку, декомпозиція першого рівня, декомпозиція під процесу аналіз збитків від викиду НХР, діаграма варіантів використання, архітектура додатку, засоби реалізації, демонстрація додатку, висновки, оприлюднення роботи

## 6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.10.2019

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Оформлення планування робіт	До 08.03.2020	
2.	Оформлення технічного завдання	До 15.03.2020	
3.	Проведення аналізу предметної області	До 22.03.2020	
4.	Проведення структурно-функціонально моделювання процесів	До 10.04.2020	
5.	Розробка додатку	До 11.05.2020	
6.	Тестування додатку	До 17.05.2020	
7.	Здача пояснювальної записки та файлів розробленого проекту	До 01.06.2020	

Студент \_\_\_\_\_  
(підпис)

Шишкін О.В.

Керівник роботи \_\_\_\_\_  
(підпис)

к.т.н., доц. Марченко А.В.

## РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток аналізу збитків від техногенних та природніх катастроф».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 21 найменувань, додатків. Загальний обсяг роботи – 110 сторінок, у тому числі 59 сторінок основного тексту, 2 сторінки списку використаних джерел, 51 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку аналізу збитків від техногенних та природніх катастроф.

В роботі проведено огляд останніх досліджень та публікацій, які стосуються аналізу збитків, аналіз аналогів розроблюваного додатку, моделювання та проектування додатку.

У роботі виконано розробку додатку, а саме таких його частин:

- модуль підрахунку реципієнтів;
- модуль роботи з мапою;
- модуль роботи з API погоди;
- модуль розрахунку збитків від викиду НХР;
- модуль розрахунку збитків від лісової пожежі;
- модуль розрахунку збитків від землетрусі;
- модуль модератора системи;
- модуль адміністратора системи.

Результатом проведеної роботи є розроблений web-додаток аналізу збитків від техногенних та природніх катастроф.

Ключові слова: катастрофа, web-додаток, збитки, мапа, реципієнти, аналіз.

## ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	8
1.1 Огляд останніх досліджень і публікацій .....	8
1.2 Аналіз програмних продуктів - аналогів.....	9
1.3 Постановка задачі .....	11
2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ .....	13
2.1 Модельовання роботи web-додатку в IDEF0.....	13
2.2 Модельовання варіантів використання .....	20
2.3 Проектування моделі бази даних .....	22
3 РОЗРОБКА WEB-ДОДАТКУ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ .....	25
3.1 Архітектура програмного додатку.....	25
3.2 Програмна реалізація .....	27
3.3 Використання програмного додатку .....	43
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	61
ДОДАТОК А.....	63
ДОДАТОК Б.....	76
ДОДАТОК В .....	87
ДОДАТОК Г.....	88

## ВСТУП

У сучасному світі кожного дня відбуваються природні або техногенні катастрофи. Їх масштаби варіюються від локальних до трансграничних [1], але якого б масштабу не була катастрофа, все одно необхідно проводити аналіз збитків для того, щоб визначити зони враження, матеріальні та людські збитки. На момент виникнення катастрофи аналіз повинен бути проведений якомога швидше та точніше, тому актуальною проблемою, яку необхідно вирішити, є розробка web-додатку, який би виконував дані розрахунки.

Об'єктом дослідження дипломного проектування є процес розрахунку збитків від катастрофи природного або техногенного характеру.

Предметом дослідження є web-додаток аналізу збитків від техногенних та природних катастроф.

Метою кваліфікаційної роботи бакалавра є розроблення web-додатку, що буде використовуватися зацікавленими особами, для аналізу збитків від техногенних або природних катастроф.

Для досягнення мети кваліфікаційної роботи потрібно вирішити такі задачі:

- аналіз існуючих методик визначення зони враження від природних або техногенних катастроф;
- дослідження існуючих методів аналізу збитків від техногенних та природних катастроф;
- дослідження існуючих програмних рішень аналізу збитків від техногенних та природних катастроф.

Причиною ініціалізації проекту є потреба в модернізації та розширенні функціоналу вже існуючого рішення, додання нових типів катастроф для яких буду проводитися аналіз збитків. Розроблений продукт може бути також використаний для подальшого користування.

Результатом проекту буде web-додаток, що надаватиме змогу потенційним користувачам швидко та точно провести аналіз збитків від техногенних та природніх катастроф, а також рівень задоволення керівника дипломної роботи та комісії, що стосується реалізації студентом поставленої задачі із супроводжуючою документацією, оформленою як пояснювальна записка.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень і публікацій

Надзвичайні ситуації, аварії, катастрофи супроводжують людство протягом всієї історії його існування. Умовно, їх можна поділити на 2 види катастроф: природнього та техногенного характеру. До катастроф природнього характеру можна віднести: повені, лісові пожежі, землетруси, урагани та інше. До катастроф техногенного характеру відносять: радіаційні аварії, викиди небезпечних хімічних речовин (НХР), пожежі та вибухи в промислових або житлових зонах, та інше.

В ході виконання бакалаврської кваліфікаційної роботи, буде розроблено web-додаток для аналізу збитків від природніх або техногенних катастроф. Природні катастрофи, які будуть представлені в додатку: лісові пожежі та землетруси, катастрофа техногенного характеру, що буде розглянута – викиди НХР.

Для виконання аналізу збитків від перелічених катастроф, існують методики, в яких описано процес визначення зон враження, та на основі інформації про території, що потрапили в ці зони, провести аналіз збитків які були нанесено.

За основу аналізу збитків від викиду НХР, було взято методику, яка представлена в дисертаційній роботі к.е.н., старшого викладача кафедри управління факультету економіки та менеджменту Сумського державного університету Опанасюк Юлії [6]. Розрахунки виконуються базуючись на типі речовини, яка була викинута, погодних умовах та місцевості в якій відбувся викид. Алгоритм виконання розрахунку наступний:



1. Визначення глибини прогнозованої зони хімічного забруднення, Гпзхз, км.
2. Визначення ширини прогнозованої зони хімічного забруднення, Шпзхз, км.
3. Визначення площі прогнозованої зони забруднення, Spзхз, км<sup>2</sup>.
4. Визначення площі зони можливого хімічного забруднення Sзмхз, км<sup>2</sup>.
5. Нанесення зони враження на мапу.
6. Враховуючи об'єкти які потрапили до зони враження визначаються збитки які були нанесені катастрофою.

Для аналізу збитків. як від лісової пожежі так і від землетрусу, за основу було взято «Методику визначення розміру і площі зон руйнувань в осередку землетрусу» [3] та «Методику розрахунку параметрів лісових пожеж» [7]. В обох випадках алгоритм розрахунку схожий і представлений наступними кроками:

1. Визначити зони враження.
2. Нанесення зони враження на мапу.
3. Враховуючи об'єкти, які потрапили до зони враження визначаються збитки, які були нанесені катастрофою.

## **1.2 Аналіз програмних продуктів - аналогів**

Програмні додатки зі схожим функціоналом існують. Для аналізу було взято продукт, модернізація якого виконується – Інформаційна система обліку збитків техногенних катастроф (рис 1.1).

Інформаційна система  
обліку збитків  
техногенних катастроф



Рисунок 1.1 – Інформаційна система обліку збитків техногенних катастроф

Провівши аналіз існуючої інформаційної системи, було виявлено ряд недоліків:

- застарілий дизайн;
- виконання розрахунків на декількох статичних мапах;
- складність додавання потенційного об'єкту враження;
- ручний підрахунок реципієнтів.

Порівняння існуючої інформаційної системи з розроблюваним web-додатком аналізу збитків від техногенних та природних катастроф наведено у таблиці 1.1.

Таблиця 1.1 – Аналіз існуючого та розроблюваного додатку для аналізу збитків

Критерій	Існуючий додаток	Розроблюваний додаток
Робота з динамічною мапою	-	+
Аналіз збитків від техногенних катастроф	+	+
Аналіз збитків від природніх катастроф	-	+
Просте розширення списку аварійно-небезпечних зон	-	+
Отримання інформації про погодні умови від API	-	+
Зручний вивід звіту про розрахунки	+	+
Автоматичний підрахунок реципієнтів	-	+

### 1.3 Постановка задачі

Метою виконання даної роботи є розробка web-додатку для аналізу збитків після техногенних або природніх катастроф. Web-додаток повинен виконувати ряд задач:

- оцінка становища при аварії (визначення напрямку руху та розмірів зони розповсюдження хмари зараженого повітря, тривалості дії факторів забруднення);

- оцінка становища під час виникнення катастроф природнього характеру;
- графічне відображення результатів аналізу та визначення даних щодо ушкодження природніх ресурсів зараженої місцевості;
- розрахунок матеріальних збитків в результаті катастрофи, термінів відновлення господарської території та природних ресурсів на основі методики, представленої в роботах.

Для реалізації поставленого завдання були виділені наступні задачі дослідження:

- аналіз аналогів;
- розробка додатку;
- розробка інтерфейсу;
- розробка модулю розрахунку;
- розробка модулю роботи з мапою;
- розробка інших модулів;
- відладка та тестування;
- оформлення супроводжуючої документації.

## 2 ПРОЕКТУВАННЯ WEB-ДОДАТКУ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ

### 2.1 Моделювання роботи web-додатку в IDEF0

Процес проектування web-додатку необхідно розпочинати з розробки контекстної діаграми IDEF0. Дане рішення пояснюється тим, що діаграми в нотації IDEF0 мають чіткі та зрозумілі твердження, що описують систему в цілому, та її взаємодію з навколишнім середовищем [2]. До контекстної діаграми входять:

- Вхід: дані, що використовуються системою для отримання результату(вихідних даних).
- Вихід: дані, які отримуються в результаті роботи системи.
- Управління: інформація, управляюча діями системи.
- Механізми: ресурси, що виконують роботу в системі.

Проаналізувавши головні елементи контекстної діаграми, для аналізу збитків від техногенних або природних катастроф, було сформовано наступний перелік даних:

- Вхід: інформація про погодні умови, інформація про катастрофу.
- Вихід: звіт про результати аналізу.
- Управління: методика аналізу збитків при викиду НХР, методика аналізу збитків при лісовій пожежі, методика аналізу збитків при землетрусі, документація до web-додатку.
- Механізми: користувач, OpenStreetMap API, Dark Sky API, web-додаток.

На основі цих даних, була розроблена контекстна діаграма, що представлена на рис. 2.1.

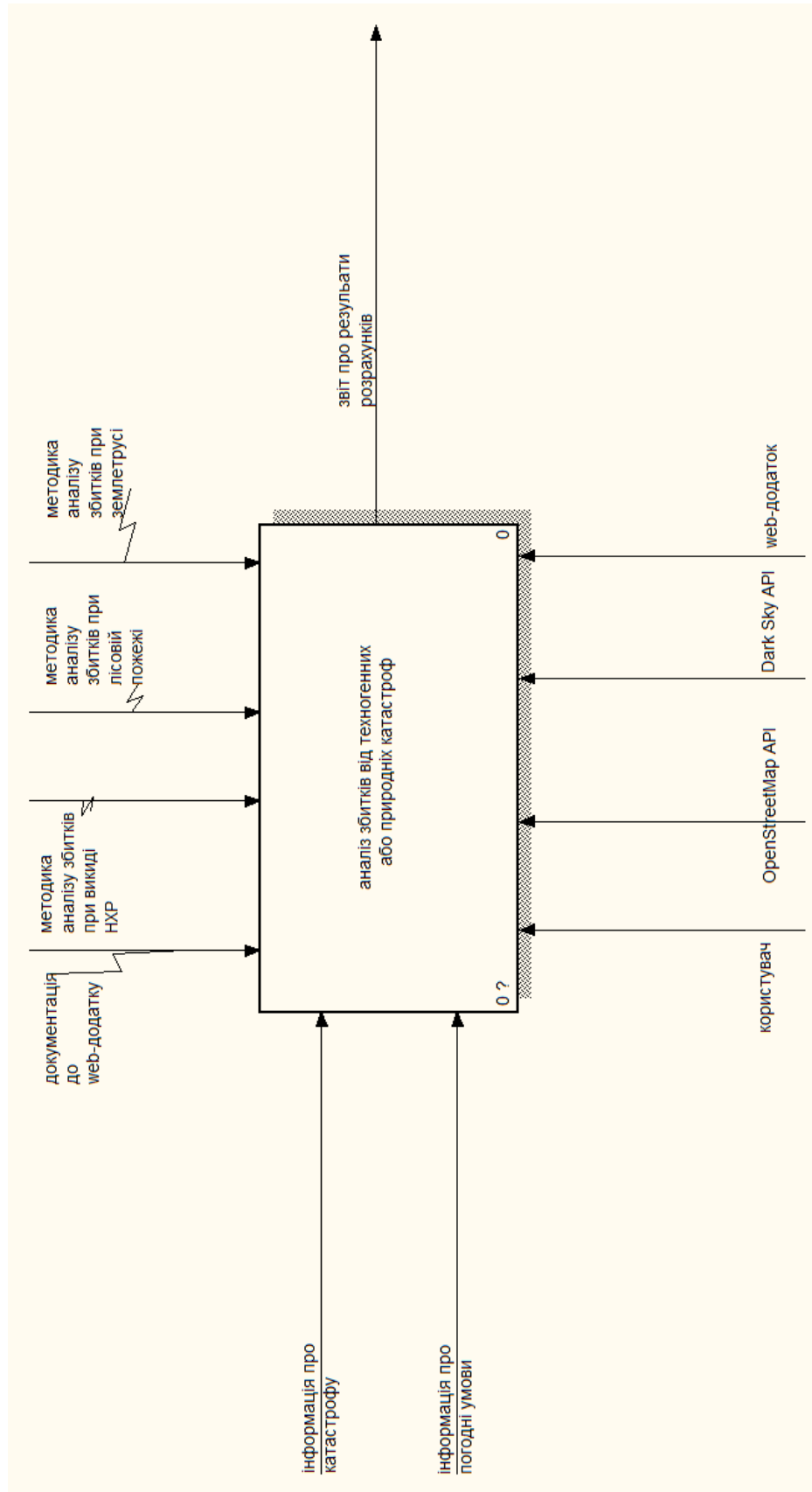


Рисунок 2.1 - Контекстна діаграма

Проаналізувавши отриману діаграму, можна отримати лише загальне уявлення про роботу системи, тому для більш детального опису логіки та послідовності робіт необхідно виконати декомпозицію. Процес аналізу збитків від техногенних або природних катастроф, розбивається на 5 під-процесів: визначення типу катастрофи, аналіз збитків від викиду НХР, аналіз збитків від лісових пожеж, аналіз збитків від землетрусів, формування звіту по збиткам.

До першого під-процесу, визначення типу катастрофи, були сформовані наступні вхідні дані:

- Вхід: інформація про катастрофу.
- Вихід: тип катастрофи.
- Управління: документація до web-додатку.
- Механізми: користувач, web-додаток.

До другого під-процесу, аналіз збитків від викиду НХР, були сформовані наступні вхідні дані:

- Вхід: тип катастрофи викид НХР, інформація про погодні умови, інформація про катастрофу.
- Вихід: результати розрахунків збитків.
- Управління: методика аналізу збитків при викиду НХР.
- Механізми: OpenStreetMap API, Dark Sky API, web-додаток.

До третього під-процесу, аналіз збитків від лісових пожеж, були сформовані наступні вхідні дані:

- Вхід: тип катастрофи лісова пожежа, інформація про погодні умови, інформація про катастрофу.
- Вихід: звіт про результати аналізу.
- Управління: методика аналізу збитків при лісовій пожежі.
- Механізми: OpenStreetMap API, Dark Sky API, web-додаток.

До четвертого під-процесу, аналіз збитків землетрусів, були сформовані наступні вхідні дані:

- Вхід: тип катастрофи землетрус, інформація про погодні умови, інформація про катастрофу.

- Вихід: результати розрахунків збитків.

- Управління: методика аналізу збитків при землетрусі.

- Механізми: OpenStreetMap API, Dark Sky API, web-додаток.

До п'ятого під-процесу, формування звіту про збитки, були сформовані наступні вхідні дані:

- Вхід: результати розрахунків збитків.

- Вихід: звіт про результати аналізу.

- Управління: документація до web-додатку.

- Механізми: web-додаток.

Діаграма другого рівня представлена на рис.2.2.

Після виконання декомпозиції на чотири під-процеси, три з них також потрібно декомпонувати для більш чіткого розуміння процесів, що відбуваються під час розробки додатку, а саме: аналіз збитків від викиду НХР, аналіз збитків від лісових пожеж, аналіз збитків від землетрусів.

Процеси аналізу збитків від викиду НХР, лісових пожеж та землетрусів, мають однаковий алгоритм дій, тому їх декомпозицію можна представити на прикладі викиду НХР.

Процес аналізу збитків від викиду НХР, розбивається на 3 під-процеси: визначення розмірів зони враження, визначення територій які потрапили в зону враження, аналіз збитків.

До першого під-процесу, визначення розмірів зони враження, були сформовані наступні вхідні дані:

- Вхід: тип катастрофи викид НХР, , інформація про погодні умови, інформація про катастрофу.

- Вихід: розміри зони враження.

- Управління: методика аналізу збитків при викиду НХР.

- Механізми: Dark Sky API, web-додаток .



До другого під-процесу, визначення територій, які потрапили в зону враження, були сформовані наступні вхідні дані:

- Вхід: розміри зони враження, інформація про погодні умови, інформація про катастрофу.

- Вихід: території які потрапили в зону враження, інформація про погодні умови, інформація про катастрофу.

- Управління: методика аналізу збитків при викиду НХР.

- Механізми: OpenStreetMap API, web-додаток.

До третього під-процесу, аналіз збитків, були сформовані наступні вхідні дані:

- Вхід: території які потрапили в зону враження, інформація про катастрофу.

- Вихід: звіт про результати аналізу.

- Управління: методика аналізу збитків при викиду НХР.

- Механізми: web-додаток.

Декомпозиція процесів аналізу збитків представлені на рис. 2.3.

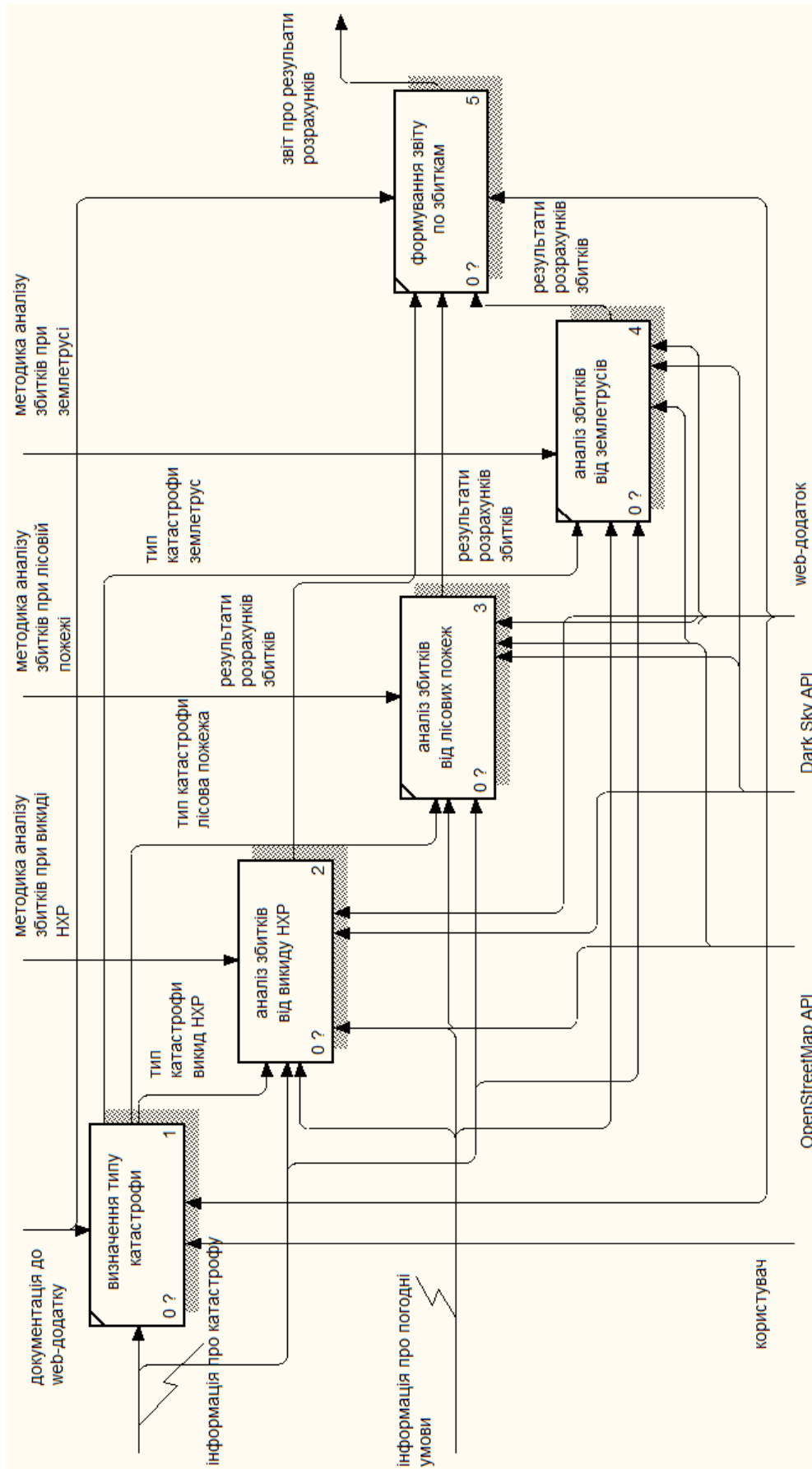


Рисунок 2.2 – Декомпозиція діаграми А-0

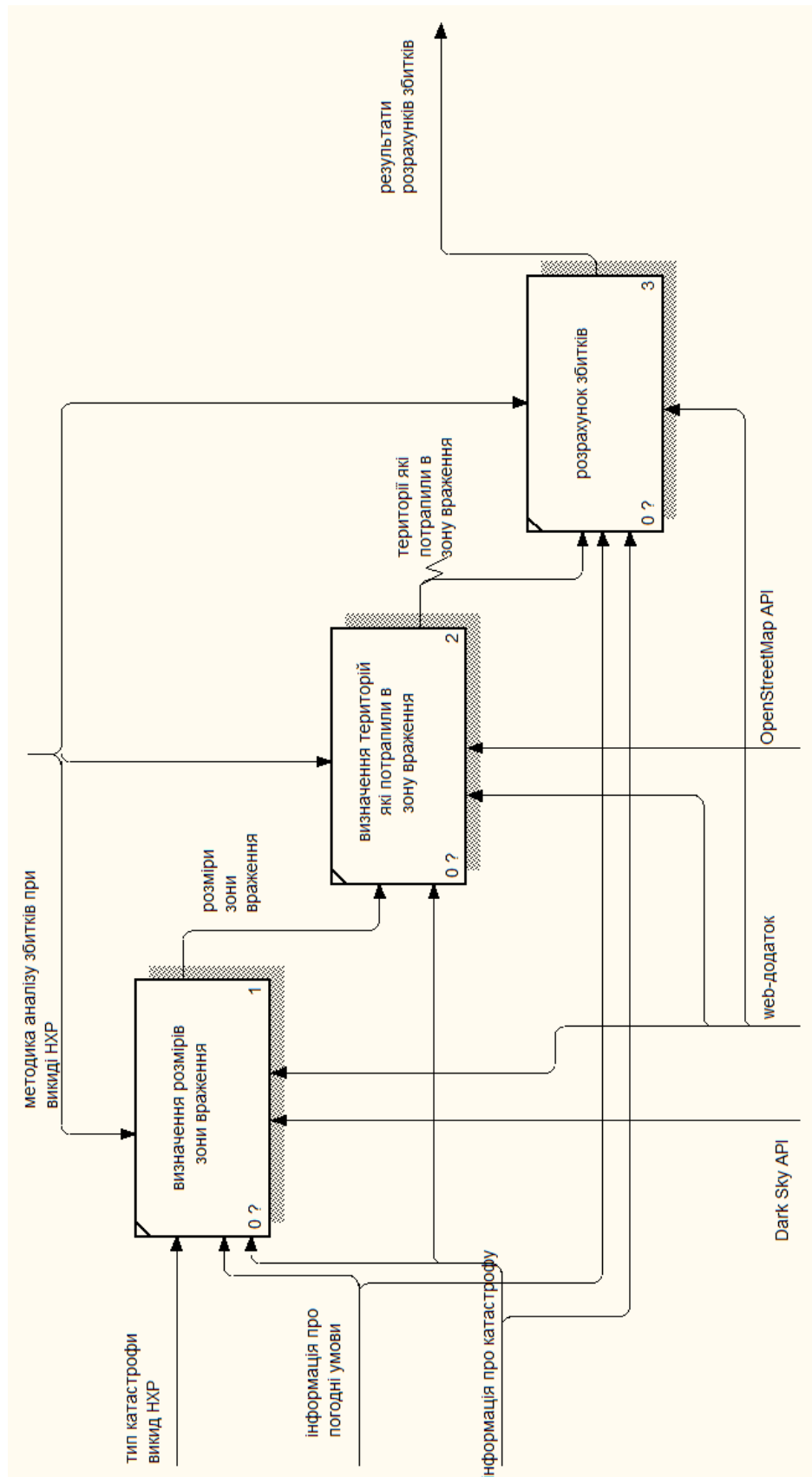


Рисунок 2.3 – Декомпозиція процесу аналізу збитків при викиду НХР

## 2.2 Моделювання варіантів використання

Наступним кроком після розробки контекстних діаграм, необхідно розробити діаграми варіантів використання (Use Case Diagram). Дана діаграма відображає взаємодію акторів та варіантів використання. Варіанти використання – це опис дій, які система може здійснювати у відповідь на зовнішні дії акторів. Розробка даного типу діаграм необхідна для того, щоб показати інформацію про те, які функції виконує система і яким чином кожен актор взаємодіє з нею [4].

Для розроблюваної системи було визначено наступних акторів:

- Відвідувач – має доступ до функціоналу виконання аналізу збитків.
- Користувач з правом читання-запису – має доступ до функціоналу виконання аналізу збитків та вносити інформацію про аварійно небезпечні об'єкти.
- Адміністратор – має доступ до розрахунків та повного налаштування додатку.
- MySQL – зберігає інформацію.
- Dark Sky API – надає інформацію про погодні умови.
- OpenStreetMap – надає доступ до мапи.

Після визначення акторів в системі, формується перелік варіантів використання. Варіанти використання:

- Виконання аналізу збитків від викиду НХР.
- Виконання аналізу збитків від лісових пожеж.
- Виконання аналізу збитків від землетрусів.
- Внесення інформації про аварійно небезпечні об'єкти.
- Налаштування web-додатку.
- Авторизація.

На основі сформованих даних про варіанти використання та акторів, було розроблено Use Case діаграму, яка представлена рис. 2.4

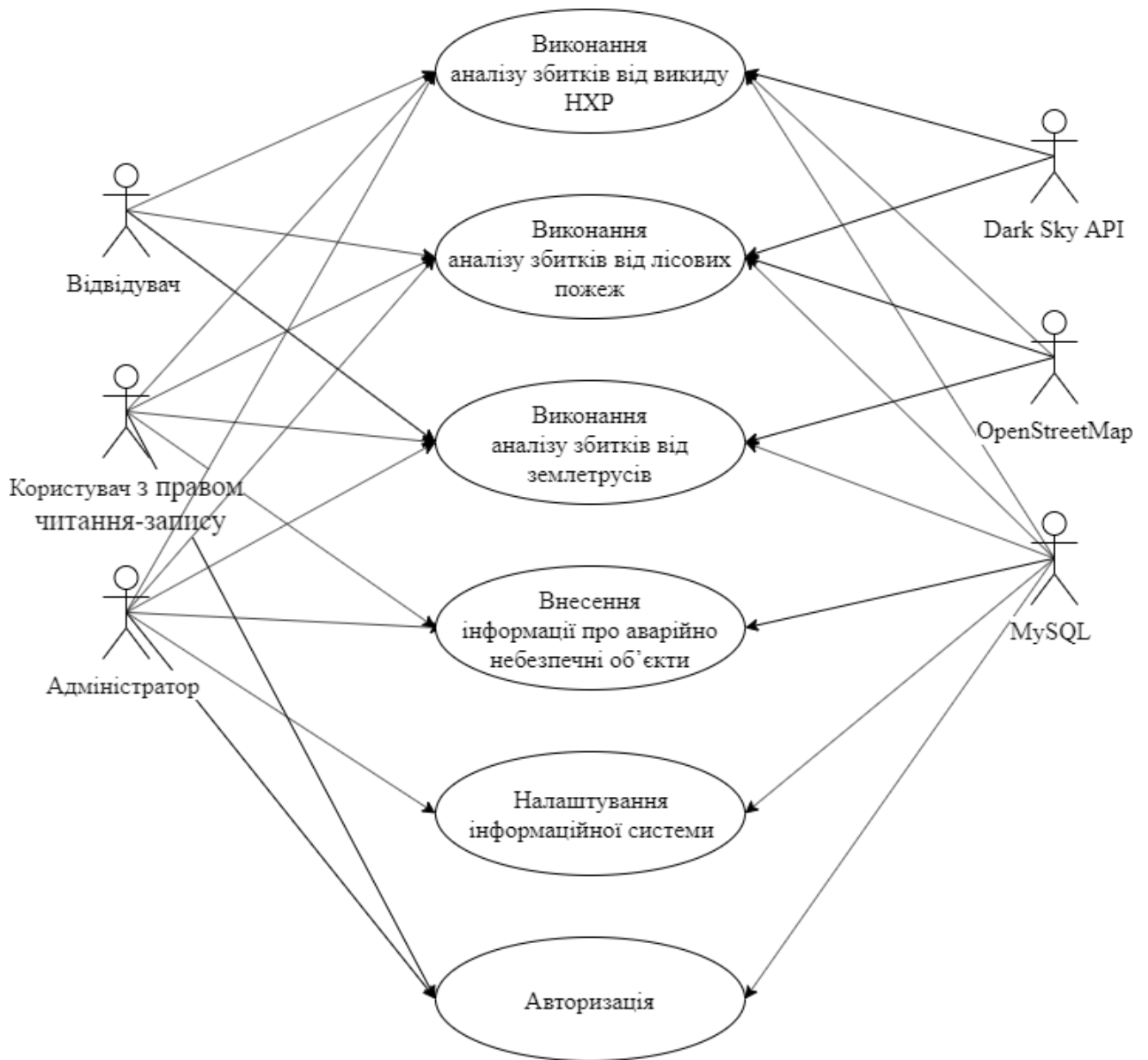


Рисунок 2.4 – Use Case Diagram

### 2.3 Проектування моделі бази даних

Будь-який додаток, який працює з даними, повинен зберігати дані з можливістю швидкого доступу до них для читання, редагування, видалення та додавання нових даних. Для задоволення цих потреб використовуються бази даних, якщо дані логічно пов'язані, або інформаційні бази для даних, які необхідні для роботи додатку, але не пов'язані між собою.

Розроблюваний додаток містить велику кількість даних, необхідних для аналізу збитків, які не пов'язані один з одним, тому було спроектовано інформаційну базу яка зберігає їх.

Інформаційна база буде містити 15 сутностей:

1. agriculture - буде містити інформацію про те, які сільськогосподарські культури переважають у якій області України;
2. earthquakeeffects - буде містити інформацію про наслідки землетрусу в залежності від його інтенсивності;
3. earthquake\_life\_support\_system - буде містити інформацію про те, яких наслідків землетрус нанесе системам життєзабезпечення;
4. earthquakedeath - буде містити інформацію про смерті людей, в залежності від типу будівлі, в якій вони знаходилися під час землетрусу та його інтенсивності;
5. factory - буде містити інформацію про хімічно небезпечні заводи;
6. koef\_kzm - буде містити інформацію про коефіцієнт  $K_{zm}$  необхідний для розрахунку глибини зони враження хмари НХР;
7. nhr\_cloud - буде містити інформацію про коефіцієнти зменшення глибини розповсюдження хмари НХР;
8. pollution\_duration - буде містити інформацію про тривалість дії НХР визначається;
9. population\_loss - буде містити інформацію про втрати людей під час викиду НХР, в залежності від забезпеченості людей протигазами;

10. tabledeptvalue- буде містити інформацію про глибину розповсюдження хмари забрудненого повітря у разі аварії на хімічно небезпечних об'єктах;

11. users - буде містити інформацію про модераторів та адміністраторів в системі;

12. vertical\_stability - буде містити інформацію про вертикальну стабільність повітря;

13. v\_transfer - буде містити інформацію про швидкість переносу повітряних мас;

14. wood\_damage - буде містити інформацію про збиток нанесений деревам пожежею;

15. wood\_damage\_char - буде містити опис кожного виду враження дерев;

По сформованому переліку сутностей до кожної з них було виділено їх атрибути та розроблено модель інформаційної бази, яку можна побачити на рисунку 2.5.

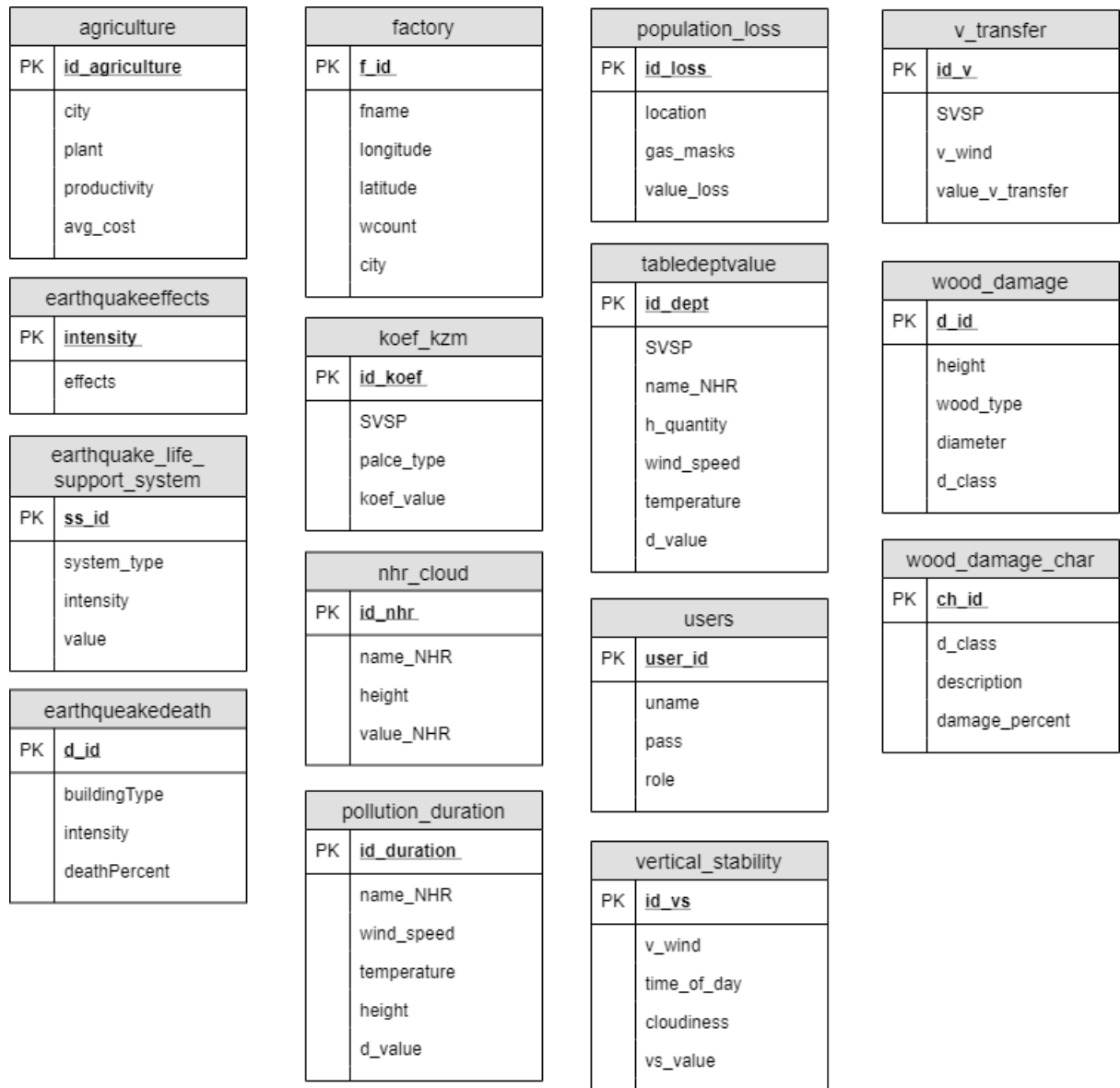


Рисунок 2.5 – Модель інформаційної бази



## 3 РОЗРОБКА WEB-ДОДАТКУ АНАЛІЗУ ЗБИТКІВ ВІД ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ

### 3.1 Архітектура програмного додатку

Перед тим як почати розробку додатку, необхідно спроектувати його архітектуру, а саме: обрати тип архітектури, структурні елементи системи, їх інтерфейси та поєднання один з одним.

Було вирішено будувати монолітний додаток, це означає, що управління всією системою буде виконуватися з одного й того самого модуля. Додаток включає в себе інформаційну базу, яка зберігає інформацію про користувачів, а також дані необхідні для розрахунків, клієнтську частину, яка відповідає за взаємодію клієнта з додатком через графічний інтерфейс, зв'язок з сторонніми API, виконання аналізу та відображення їх результатів у зручній для користувача формі, серверну частину, яка відповідає за зв'язок клієнтської частини з інформаційною базою та генерацію даних необхідних для розрахунків. Поряд з модулями, які працюють в середині додатку, він має зв'язок з двома сторонніми API, а саме Dark Sky API та Open Street Map. Dark Sky API надає інформацію про погодні умови, які необхідні для розрахунків. Open Street Map дозволяє працювати з мапою для відображення на ній зон враження, а також отримання інформації про місце виникнення катастрофи, що також необхідно для виконання розрахунків.

На рисунку 3.1 представлена діаграма компонентів, яка відображає взаємодію модулів між собою, а на рисунку 3.2 представлена схема архітектури додатку.

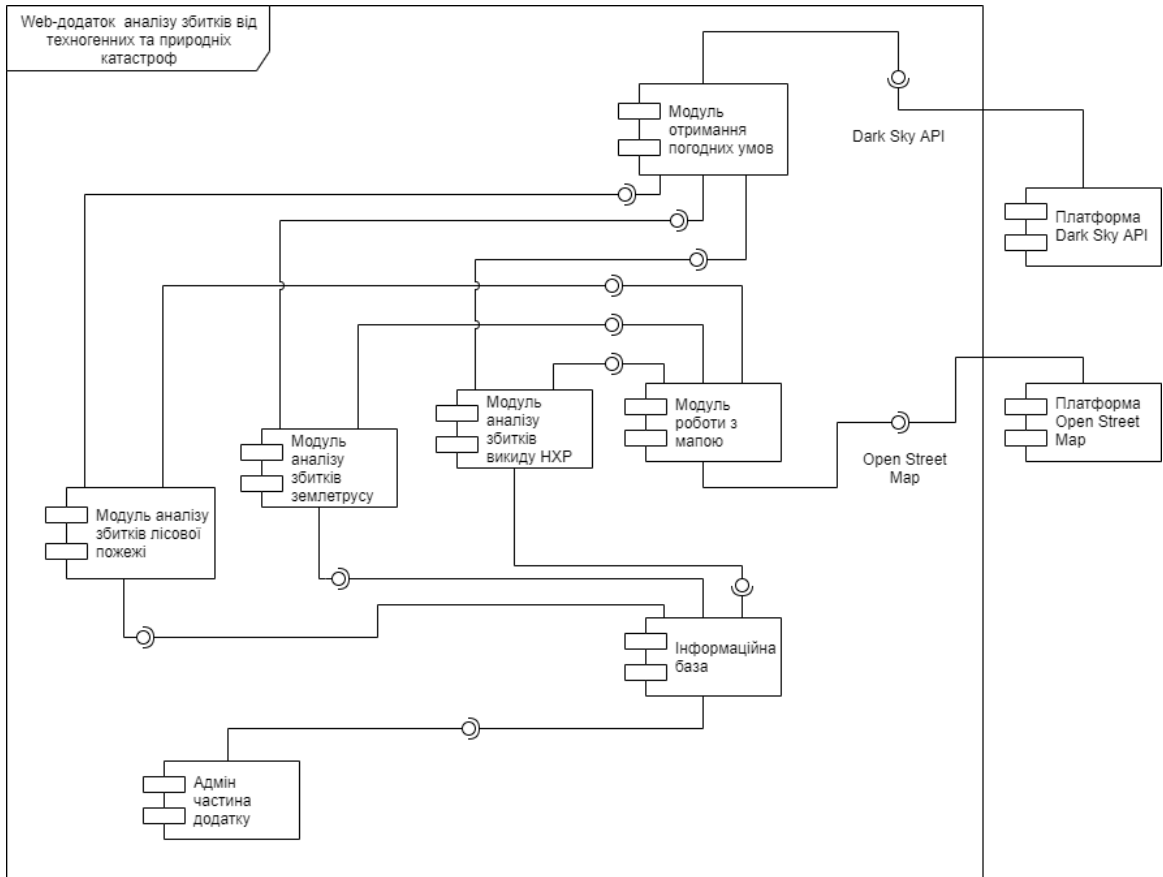


Рисунок 3.1 - Діаграма компонентів

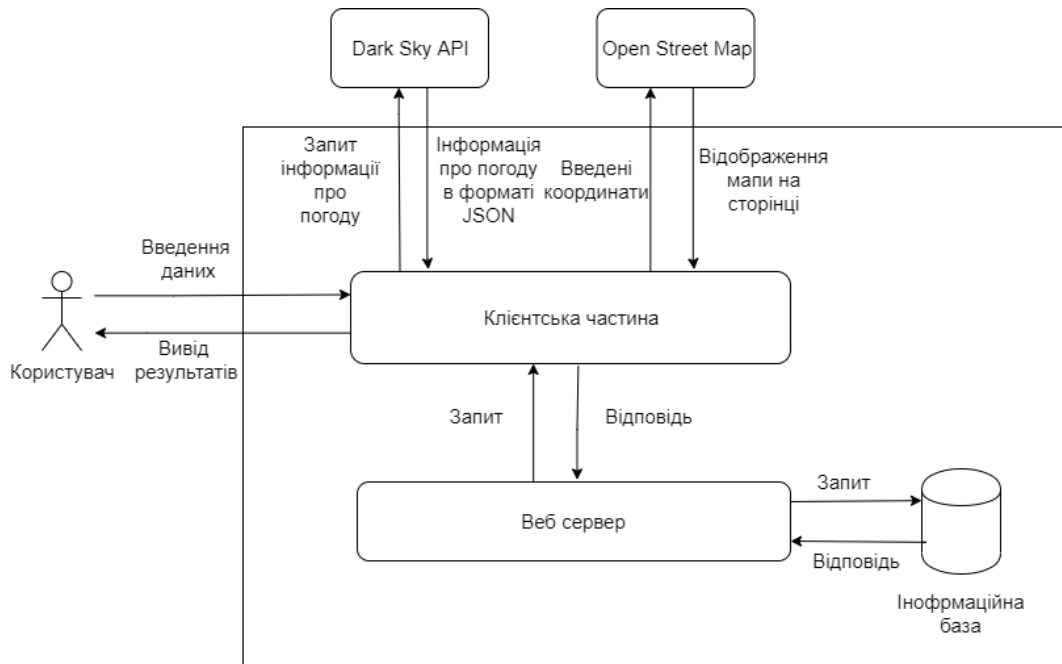


Рисунок 3.2 – Архітектура додатку

## 3.2 Програмна реалізація

Переглянути весь програмний код можна в додатку В або в гітхаб репозиторії за посиланням <https://github.com/alexe4540/diploma>.

### 3.2.1 Розробка інформаційної бази

Інформаційна база додатку була розроблена з використанням СУБД MySQL. База являє собою набір не пов'язаних між собою таблиць з даними. Кожна таблиця містить поля, тип яких залежить від даних, які в ній зберігаються.

Інформаційна база «diploma» містить 15 таблиць:

1. Таблиця «agriculture» містить інформацію про те, які сільськогосподарські культури переважають у якій області України:
  - id\_agriculture – ідентифікатор (int) {PK};
  - city – назва міста (varchar);
  - plant – назва культури (varchar);
  - productivity – очікувана прогностична врожайність (float);
  - avg\_cost – прогностична середня оптова ціна (float).
2. Таблиця «earthquakeeffects» містить інформацію про наслідки інтенсивності землетрусу:
  - intensity – значення інтенсивності (int) {PK};
  - effects – яких збитків нанесе землетрус з такою інтенсивністю (text).
3. Таблиця «earthquake\_life\_support\_system» містить інформацію про те, яких наслідків землетрус нанесе системам життєзабезпечення:
  - ss\_id – ідентифікатор (int) {PK};
  - system\_type – тип системи (enum(«водопостачання», «електропостачання», «газопостачання», «теплопостачання», «транспорт», «каналізація», «зв'язок»));

- intensity – значення інтенсивності (int);
- value – наслідки (varchar).

4. Таблиця «earthquakedeath» містить інформацію про смерті людей в залежності від типу будівлі в якій вони знаходилися під час землетрусу та його інтенсивності:

- d\_id – ідентифікатор (int) {PK};
- buildingType – тип будівлі (enum(«Дерев'яні», «Цегляні малоповерхові (1-2 поверху)», «Цегляні з неповною каркасною стіною», «Цегляні багатоповерхові»));
- intensity – значення інтенсивності (int);
- deathPercent – кількість загиблих у відсотках (int).

5. Таблиця «factory» містить інформацію про хімічно небезпечні заводи:

- f\_id – ідентифікатор (int) {PK};
- fname – назва заводу (varchar);
- longitude – широта на якій розміщений завод (double);
- latitude – довгота на якій розміщений завод (double);
- wcount – кількість працівників на заводі (float);
- city – місто в якому розташовано завод (varchar);

6. Таблиця «kof\_kzm» містить інформацію про коефіцієнт Kzm необхідний для розрахунку глибини зони враження хмари НХР:

- id\_koef – ідентифікатор (int) {PK};
- SVSP – значення вертикальної стійкості повітря (enum(«інверсія», «ізотерія», «конвекція»));
- palce\_type – місцевість на якій виникла катастрофа (enum(«Місто», «Ліс», «Село»));
- koef\_value – значення коефіцієнту (float).

7. Таблиця «nhr\_cloud» містить інформацію про коефіцієнти зменшення глибини розповсюдження хмари НХР:

- id\_nhr – ідентифікатор (int) {PK};

- name\_NHR – вид НХР (enum(«хлор», «аміак», «сірчаний ангідрид», «сірководень», «соляна кислота», «хлорпікрин», «формальдегід»));
- height – висота обвалування (enum(«1», «2», «3»));
- value\_NHR – значення коефіцієнту (float).

8. Таблиця «pollution\_duration» містить інформацію про тривалість дії НХР визначається:

- id\_duration – ідентифікатор (int) {PK};
- name\_NHR – вид НХР (enum(«хлор», «аміак», «сірчаний ангідрид», «сірководень», «соляна кислота», «хлорпікрин», «формальдегід»));
- wind\_speed – швидкість вітру (enum(«1», «2», «3», «4», «5», «10»));
- temperature – температура повітря (enum(«-20», «0», «20»));
- height – висота піддону (enum(«1», «3»));
- d\_value – значення тривалості випаровування (float).

9. Таблиця «population\_loss» містить інформацію про втрати людей під час викиду НХР в залежності від забезпеченості людей проти газами:

- id\_loss – ідентифікатор (int) {PK};
- location – вид місцевості (enum(«відкрита», «закрита»));
- gas\_masks – відсоток забезпеченості людей проти газами (enum(«0», «20», «30», «40», «50», «60», «70», «80», «90», «100»));
- value\_loss – значення втрат людей у відсотках (int).

10. Таблиця «tabledeptvalue» містить інформацію про глибину розповсюдження хмари забрудненого повітря у разі аварії на хімічно небезпечних об'єктах:

- id\_dept – ідентифікатор (int) {PK};
- SVSP – значення вертикальної стійкості повітря (enum(«інверсія», «ізотермія», «конвекція»));
- name\_NHR – вид НХР (enum(«хлор», «аміак», «сірчаний ангідрид», «сірководень», «соляна кислота», «хлорпікрин», «формальдегід»));
- h\_quantity – кількість викинутої речовини (enum(«0,5», «1», «3», «5», «10», «20», «30», «50», «70», «100», «300»));

- wind\_speed – швидкість вітру (enum(«1», «2», «3», «4», «5», «10»));
- temperature – температура повітря (enum(«-20», «0», «20»));
- d\_value – табличне значення глибини зони(float).

11. Таблиця «users» містить інформацію про модераторів та адміністраторів в системі:

- user\_id – ідентифікатор (int) {PK};
- uname – логін користувача (varchar);
- pass – пароль користувача у вигляді хеш суми (varchar);
- role – роль користувача (enum(«admin», «moderator»)).

12. Таблиця «vertical\_stability» містить інформацію про вертикальну стабільність повітря:

- id\_vs – ідентифікатор (int) {PK};
- v\_wind – швидкість вітру (enum(«0», «0.5», «2», «4»));
- time\_of\_day – період дня (enum(«Ніч», «День»));
- cloudiness – хмарність (enum(«Ясно», «Напівясно», «Хмарно»));
- vs\_value – значення вертикальної стабільності повітря (enum(«інверсія», «ізотерія», «конвекція»)).

13. Таблиця «v\_transfer» містить інформацію про швидкість переносу повітряних мас:

- id\_v – ідентифікатор (int) {PK};
- SVSP – значення вертикальної стійкості повітря (enum(«інверсія», «ізотерія», «конвекція»));
- v\_wind – швидкість вітру (enum(«1», «2», «3», «4»));
- value\_v\_transfer – значення швидкості перенесення повітряних мас (float).

14. Таблиця «wood\_damage» містить інформацію про збиток нанесений деревам пожежею:

- d\_id – ідентифікатор (int) {PK};
- height – висота нагару (enum(«0.1 - 0.5», «0.6 - 1.0», «1.0 - 1.5», «1.6 - 2.0», «2.1 - 3.0», «3.1 - 4.0», «4.1 - 5.0», «5.1 та більше»));

- wood\_type – тип лісу (enum(«1», «2»));
- diameter – середній діаметр стовбуру дерева (enum(«8», «12», «16», «20», «24», «28», «32», «36», «40»));
- d\_class – клас збитків нанесених деревам(enum(«1», «2», «3»)).

15. Таблиця «wood\_damage\_char» містить опис кожного виду враження дерев:

- ch\_id – ідентифікатор (int) {PK};
- d\_class – клас збитків нанесених деревам(enum(«1», «2», «3»)).
- description – опис (text);
- damage\_percent – відсоток дерев непригідних до реалізації.

Переглянути фізичне відображення створеної інформаційної бази можна на рисунку 3.3.

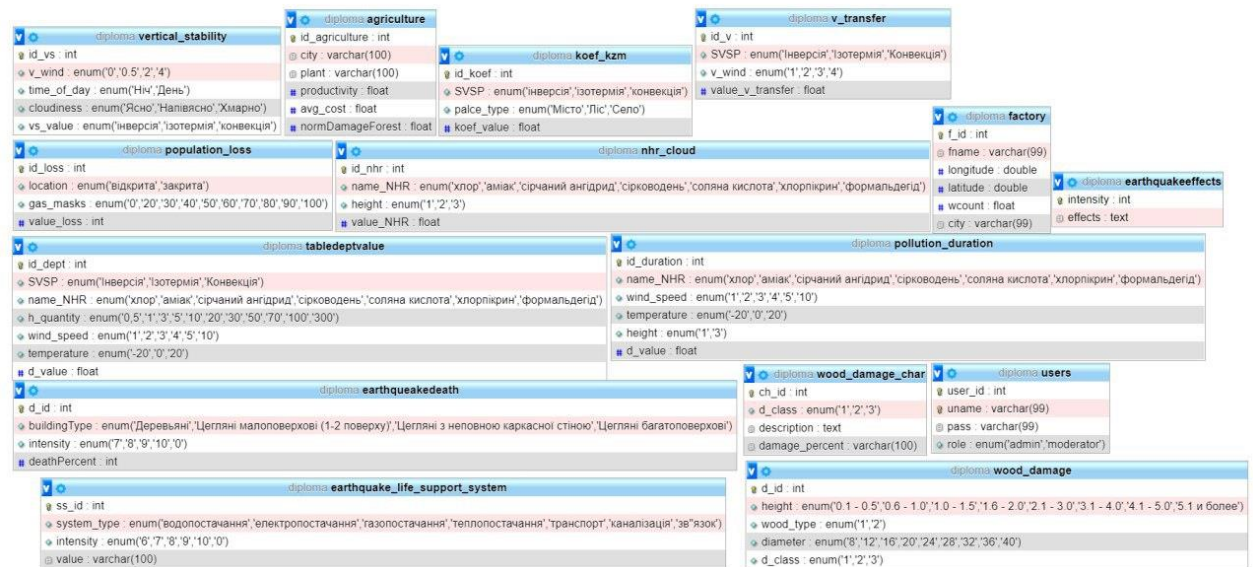


Рисунок 3.3 – Відображення фізичної реалізації інформаційної бази у вікні дизайнера phpMyAdmin

### 3.2.2 Розробка модулю підрахунку реципієнтів, які потрапили до зони враження

Важливою частиною аналізу збитків від викиду НХР, є підрахунок реципієнтів, які потрапили в зону враження.

Даний модуль працює на стороні клієнта і виконується за рахунок його обчислювальних потужностей. Модуль реалізовано мовою програмування JavaScript і він включає в себе 3 функції:

- `getData()`;
- `unitArrayToRegularArray(unitArray)`;
- `findPixelsInZone(data, zoneColor, pixelColor)`.

Функція `getData()`(рис 3.4) зв'язується з елементом графічного інтерфейсу `canvas`, отримує його контекст. З контексту, за допомогою влаштованого методу `getImageData()`, зчитується інформація про зображення у вигляді об'єкта `ImageData`, який містить інформацію про кожен піксель, вона зберігається у властивості – `data` даного об'єкта. `Data` представляє собою одномірний масив, у якому кожен чотири елементи відповідають за колір одного пікселя. Перший із чотирьох елементів відповідає за частку червоного пікселя, другий – зеленого, третій – синього, четвертий – альфа-канал. З функції повертається масив `arrayOfRGBAValue`.

```
function getData() {  
    let c = document.getElementById("canvas");  
    let ctx = c.getContext("2d");  
    let data = ctx.getImageData(0, 0, c.width, c.height)  
  
    let arrayOfRGBAValue = unitArrayToRegularArray(data);  
    return arrayOfRGBAValue;  
}
```

Рисунок 3.4 – Функція `getData`



Для визначення кількості реципієнтів даний масив даних є незручним, тому в функції `getData()` викликається допоміжна функція модулю – `unitArrayToRegularArray(unitArray)`(рис 3.5).

Функція `unitArrayToRegularArray(unitArray)` приймає на свій вхід об'єкт `ImageData`. В ході виконання функції з властивості масиву `data` об'єкту `ImageData` формується новий масив `rgbaArray`, який представляє собою двомірний масив, що має кількість стовпців рівну значенню, що міститься у властивості `width`, а кількість рядків відповідає значенню, що міститься у властивості `height`, об'єкту `ImageData`. Кожен елемент масиву, представляє собою масив з чотирьох елементів. Перший із чотирьох елементів відповідає за частку червоного пікселя, другий – зеленого, третій – синього, четвертий – альфа-канал. З функції повертається масив `rgbaArray`.

```
function unitArrayToRegularArray(unitArray){
  let regularArray = [];
  let rgbaArray = [];

  for(let i = 0; i < unitArray.data.length; i+=4){
    regularArray.push([unitArray.data[i],unitArray.data[i+1], unitArray.data[i+2], unitArray.data[i+3]]);
  }

  let counter = 0;

  for (let i = 0; i < unitArray.height; i++){
    let tmpArray = [];
    for (let j = 0; j < unitArray.width; j++){
      tmpArray.push(regularArray[j+counter]);
    }
    rgbaArray.push(tmpArray);
    counter += unitArray.width;
  }

  return rgbaArray;
}
```

Рисунок 3.5 – Функція `unitArrayToRegularArray`

Функція `findPixelsInZone(data, zoneColor, pixelColor)`(рис 3.6) викликається з основного потоку програми і приймає на свій вхід:

- `data` – масив пікселів у якому відбувається пошук реципієнтів;

- zoneColor – об’єкт, який містить значення rgb каналів кольору для знаходження границь зони враження.
- pixelColor – об’єкт, який містить значення rgb каналів кольору для знаходження реципієнтів у зоні.

Під час викання функції відбувається прохід по всім значенням вхідного масиву та пошук пікселя, що відповідає кольору границі зони враження. Після знаходження такого пікселя починається підрахунок пікселів, що відповідають кольору реципієнта. На вихід подається кількість реципієнтів знайдених в зоні враження. Блок схема функції findPixelsInZone представлена на рисунку 3.7.

```
function findPixelsInZone(data, zoneColor, pixelColor){
  let totalCounter = 0;
  let positionOfLastZonePixel;

  for (let row of data){
    let tmpCounter = 0;
    let countPixel = false;
    let addToTotalCount = false;
    for (let i = 0; i < row.length - 1; i++){
      if(row[i][0] === zoneColor.r && row[i][1] === zoneColor.g && row[i][2] === zoneColor.b){
        positionOfLastZonePixel = i;
      } else if(!countPixel && i - positionOfLastZonePixel === 1){
        countPixel = !countPixel;
      }

      if(countPixel){
        if(row[i][0] === pixelColor.r && row[i][1] === pixelColor.g && row[i][2] === pixelColor.b){
          tmpCounter++;
        }
        if(row[i + 1][0] === zoneColor.r && row[i + 1][1] === zoneColor.g && row[i + 1][2] === zoneColor.b){
          totalCounter += tmpCounter;
          break;
        }
      }
    }
  }

  return totalCounter;
}
```

Рисунок 3.6 – Функція findPixelsInZone

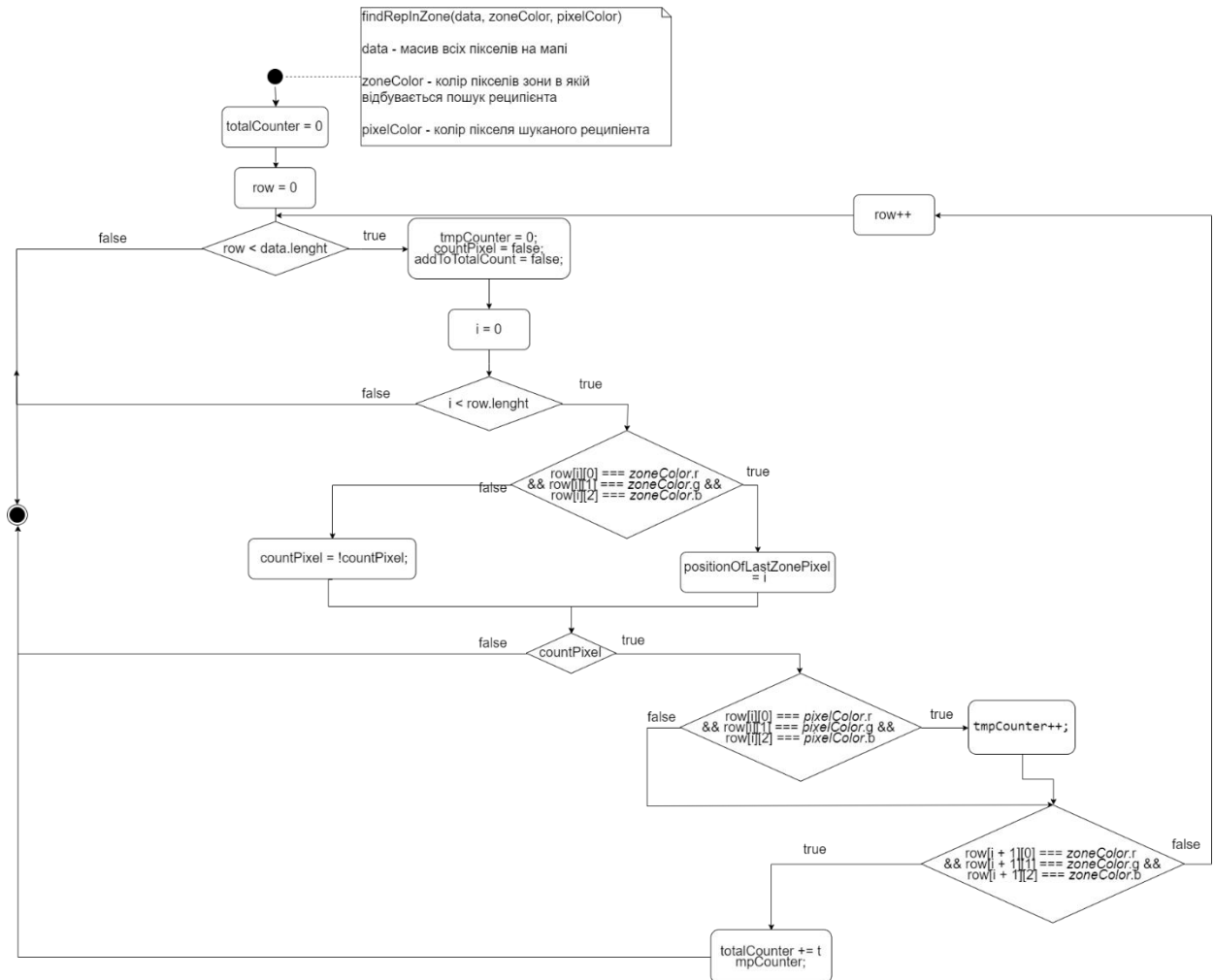


Рисунок 3.7 – Блок схема функції findPixelsInZone

### 3.2.3 Розробка модулю роботи з API погоди

Для всіх розрахунків, які виконуються в розроблюваному додатку, необхідна інформація про погодні умови. Для цього було вирішено використовувати API, яке надає інформацію про погодні умови з метеостанції. Дане рішення було прийнято тому, що залишити введення даних про погодні умов на користувача підвищить час, який буде витрачено на взаємодію з додатком, а також підвищить шанс допущення помилок або введення не вірних даних.

Для роботи було обрано Dark Sky API. Для того, щоб отримати доступ для роботи з API спочатку потрібно зареєструватися на сайті та згенерувати власний ключ для підключення.

Для отримання інформації від API, клієнтська частина додатку генерує посилання, за допомогою функції `getWetherData(latitude, longitude)`(рис 3.8). Функція приймає параметри: `latitude` – широта, `longitude` – довгота, місцевості в якій треба дізнатися погодні умови. Згенероване посилання передається до функції `getJSONWether(url)`.

```

2  const KEY = "199cb0b508acd58820ca7ed8fea3afb3";
3
4  function getWetherData(latitude, longitude){
5      const url=`https://api.darksky.net/forecast/${KEY}/${latitude},${longitude}?lang=uk&units=auto`;
6
7      return getJSONWether(url);
8  }

```

Рисунок 3.8 – Функція `getWetherData`

Функція `getJSONWether(url)`(рис. 3.9) приймає на вхід, параметр `url`, що є посиланням за яким сервер буде звертатися до API. Функція посилає асинхронний POST запит на сервер і очікує доки від нього не поступить відповідь.

```

1  async function getJSONWether(url) {
2      let result;
3
4      await $.ajax({
5          type : "POST", //type of method
6          url  : "/router/wetherRouter.php", //your page
7          data : { url : url}, // passing the values
8          success: function(res){
9              result = res;
10         }
11     });
12
13     return result;
14 }
15

```

Рисунок 3.9 – Функція `getJSONWether`

На сервері виконується перевірка, якщо надійшов POST запит згенерованим просиланням, то за допомогою влаштованої функції мови php: `file_get_contents` зчитується вміст відповіді і перетворюється в рядок, який приходить на клієнтську частину, де перетворюється в об'єкт, якій містить дані про погодні умови(рис. 3.10).

```
{
  "latitude": 42.3601,
  "longitude": -71.0589,
  "timezone": "America/New_York",
  "currently": {
    "time": 1509993277,
    "summary": "Drizzle",
    "icon": "rain",
    "nearestStormDistance": 0,
    "precipIntensity": 0.0089,
    "precipIntensityError": 0.0046,
    "precipProbability": 0.9,
    "precipType": "rain",
    "temperature": 66.1,
    "apparentTemperature": 66.31,
    "dewPoint": 60.77,
    "humidity": 0.83,
    "pressure": 1010.34,
    "windSpeed": 5.59,
    "windGust": 12.03,
    "windBearing": 246,
    "cloudCover": 0.7,
    "uvIndex": 1,
    "visibility": 9.84,
    "ozone": 267.44
  },
}
```

Рисунок 3.10 – Частина відповіді від API

### 3.2.4 Розробка модулю роботи з мапою

Головною особливістю розроблюваного додатку є робота з динамічною мапою. В якості мапи, з якою буде проводитися робота, було обрано Open Street Map, так як вона має велику спільноту людей, які кожного дня додають або оновлюють існуючі дані на ній, що забезпечує актуальність даних з якими буде виконуватися робота.

Для роботи з мапою необхідно використовувати JS бібліотеку Leaflet, яка надає функціонал переміщення, збільшення мапи, виставлення маркерів, на мапі та виконувати інші дії за допомогою JS команд.

Після завантаження сторінки, до неї одразу додається блок з ідентифікатором «map», в цей же час за допомогою методу map() бібліотеки Leaflet створюється об'єкт мапи. До неї додається заголовок з посиланням на сайт Open Street Map, що відповідає користувальницьким умовам використання даної мапи, та виконується рендер готової мапи на сторінку. Переглянути код описаного процесу можна на рисунку 3.11.

```

19  const map = L.map('map').setView([50.452554, 30.561522], 5);
20  L.tileLayer('https://{s}.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png', {
21    attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
22  }).addTo(map);
23  L.control.scale().addTo(map);

```

Рисунок 3.11 – Код додавання мапи на сторінку

Модуль складається з 2 частин. Перша частина працює безпосередньо з мапою, до неї відносяться функції workWithMap(coordinates, scale)(Рис 3.12) та goFunc()(рис. 3.13). Друга частина модулю виконує роботу вже з елементом canvas в якому знаходиться скріншот мапи, до неї відноситься функція drawOnMap(drawFuncName, color, canvas, latitude, longitude, scale, radius, angle) (рис. 3.14).

Функція workWithMap приймає два параметри: coordinates – масив координат на мапі(широта та довгота), scale – масштаб мапи. За допомогою

методу `setView` об'єкту `map`, камера мапи переміщується до вказаних координат з урахуванням заданого масштабу. Після переміщення камери, викликається функція `goFunc()`.

Функція `goFunc()` не приймає ніяких параметрів. За допомогою допоміжної функції `html2canvas(DOMObj, settingObj)`, яка в даному розділі не описується, робиться скріншот вказаного в параметрі `DOMObj` елементу, та після чого додається на сторінку до елементу `canvas`.

```

1  async function workWithMap(coordinates, scale) {
2      let latitude = coordinates.latitude;
3      let longitude = coordinates.longitude;
4
5      map.setView([latitude, longitude], scale);
6
7      let promise = new Promise((resolve, reject) => {
8          setTimeout(() => resolve("готово!"), 1000)
9      });
10     let result = await promise;
11
12     goFunc();
13
14     return new Promise((resolve, reject) => {
15         setTimeout(() => resolve("готово!"), 1000)
16     });
17 }

```

Рисунок 3.12 – Функція `workWithMap`

```

25 function goFunc() {
26     html2canvas(document.querySelector("#map"), {
27         allowTaint: true,
28         useCORS: true,
29     }).then(function (canvas) {
30         let mapImg = document.querySelector('#mapImg');
31         let cnvs = document.querySelector("canvas");
32         if (cnvs) cnvs.remove();
33         canvas.id = 'canvas';
34         mapImg.appendChild(canvas);
35     });
36 };

```

Рисунок 3.13 – Функція `goFunc`

Функція `drawOnMap` відповідає за відображення зони враження в залежності від того, які вхідні параметри було отримано, вона приймає вісім вхідних параметрів:

- `drawFuncName` – назва функції від якої залежить вид зони враження;
- `color` – об'єкт, в якому зберігаються `rgb` дані про колір границі зони;
- `canvas` – об'єкт `canvas`;
- `latitude` – довгота;
- `longitude` – широта;
- `scale` – масштаб;
- `radius` – радіус зони враження;
- `angle` – азимут вітру.

В залежності від значення параметру `drawFuncName`, викликається одна з допоміжних функцій для відображення зони враження. Може приймати одне з трьох значень: `drawArc`, `drawEllipse` та `drawFireZone`.

Значення `drawArc` відповідає за роботу із зоною враження від землетрусу та малює коло, з урахуванням масштабу та географічного положення вказаного центру кола, за заданим радіусом.

Значення `drawEllipse` відповідає за роботу із зоною враження від викиду НХР, та малює продовгуватий еліпс, який будується по пів-осям, що залежать від швидкості вітру, при малюванні еліпсу також враховується масштаб та географічне положення вказаного центру еліпсу.

Значення `drawFireZone` також малює еліпс на основі пів осей, але вони в свою чергу залежать від швидкості розповсюдження полум'я в напрямку тилу, флангів та фронту, при малюванні еліпсу також враховується масштаб та географічне положення вказаного центру еліпсу.

Результати виконання функції для різних значення параметру `drawFuncName` можна побачити на рисунках 3.15-3.17.



```

function drawOnMap(drawFuncName, color, canvas, Latitude, Longitude, scale, radius, angle) {
  let ctx = canvas.getContext('2d');

  if (drawFuncName == 'drawArc'){
    let rad = pixelValue(latitude, radius * 1000, scale)

    drawArc(color, ctx, [canvas.width / 2, canvas.height / 2], rad);
  } else if (drawFuncName == 'drawEllipse') {
    let sPixelA = pixelValue(latitude, radius[0], scale);
    let sPixelB = pixelValue(latitude, radius[1], scale);

    drawEllipse(color, ctx, [canvas.width / 2, canvas.height / 2], [sPixelA, sPixelB], angle);

    return;
  } else if (drawFuncName == 'drawFireZone'){
    let sPixelA = pixelValue(latitude, radius[0], scale);
    let sPixelB = pixelValue(latitude, radius[1], scale);

    drawFireZone(color, ctx, [canvas.width / 2, canvas.height / 2], [sPixelA, sPixelB], angle);

    return;
  }
}

```

Рисунок 3.14 – Функція drawOnMap.

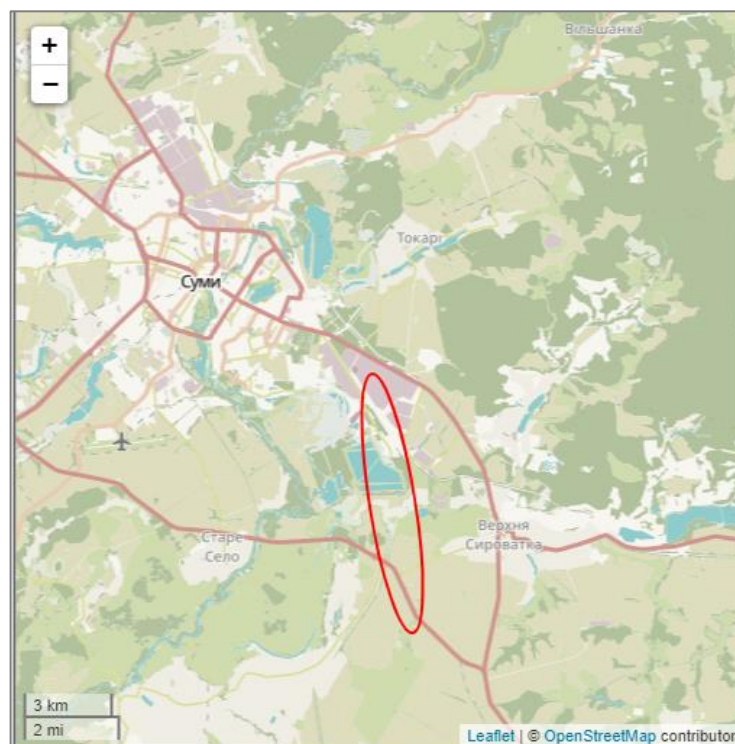


Рисунок 3.15 – Результат виконання функції drawOnMap з значенням drawEllipse параметру drawFuncName

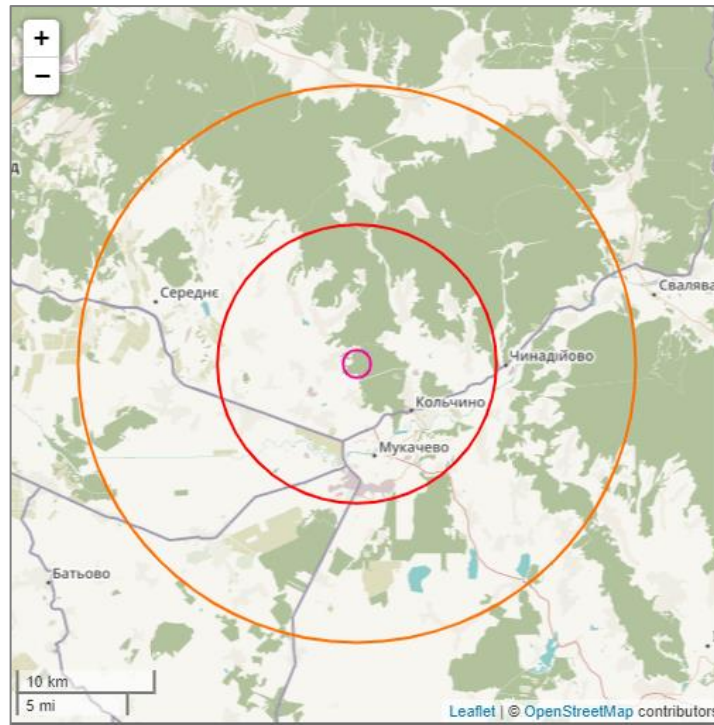


Рисунок 3.16 – Результат виконання функції drawOnMap з значенням drawArc параметру drawFuncName

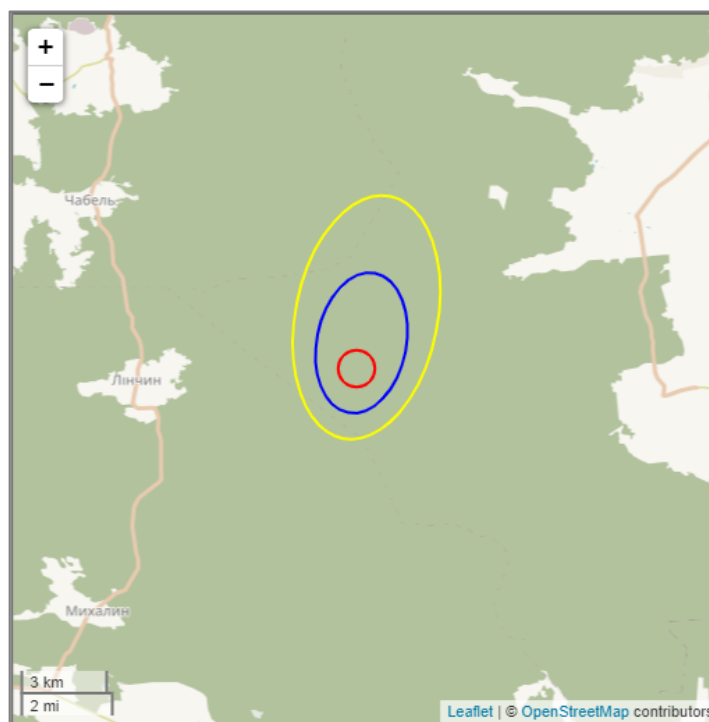


Рисунок 3.17 – Результат виконання функції drawOnMap з значенням drawFireZone параметру drawFuncName

### 3.3 Використання програмного додатку

Після того як web-додаток буде відкрито, користувач потрапить на головну сторінку, яка складається з 3х основних блоків:

- Шапка сайту – містить посилання на головну сторінку та сторінки з розрахунками по кожному із представлених видів збитків.
- Основний розділ сторінки – містить інформацію про кожний вид катастрофи, які представлено в додатку, після наведеної інформації наводиться посилання на сторінку описаної катастрофи.
- Підвал сайту – містить інформацію про розробника та кнопку входу для адміністраторів або модераторів додатку. Якщо користувач вже авторизований то замість кнопки входу відображаються посилання на модер-панель, якщо авторизований користувач має право доступу модератор. Якщо авторизований користувач має права адміна, то відображаються посилання на адмін- та модер-панелі. Незалежно від ролі користувача відображається кнопка виходу з акаунту.

Вигляд головної сторінки можна побачити на рисунку 3.18.

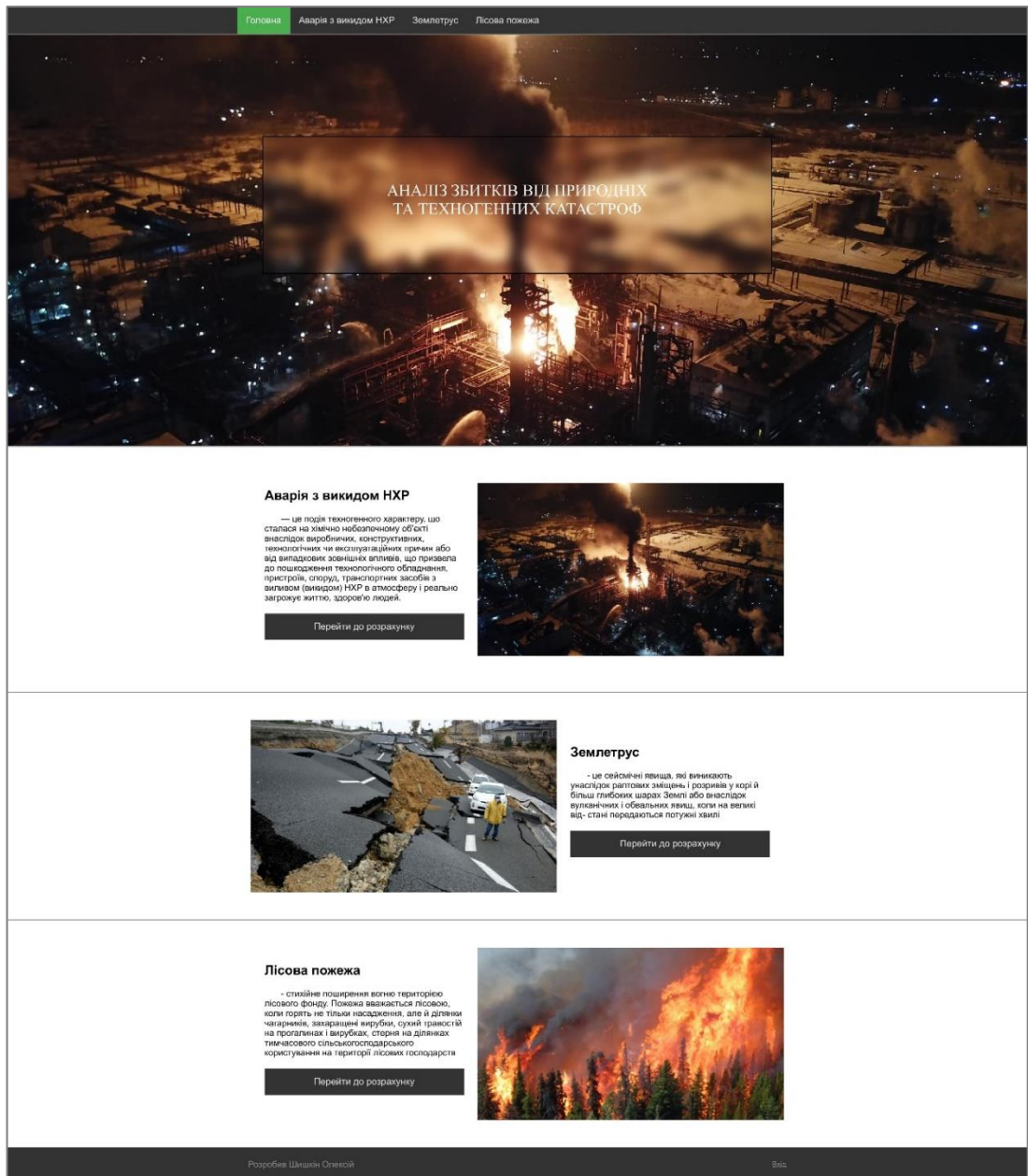
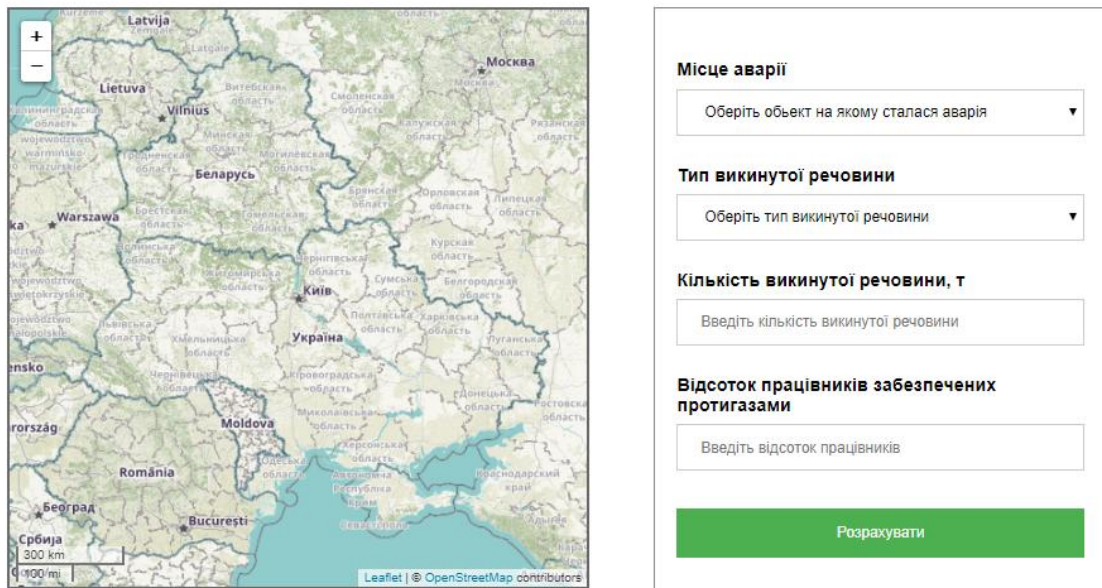


Рисунок 3.18 – Вигляд головної сторінки додатку

Перейшовши на сторінку аналізу збитків від аварії з викидом НХР(рис. 3.19), користувач бачить мапу та форму для заповнення даних.



**Місце аварії**

Оберіть об'єкт на якому сталася аварія ▼

**Тип викинутої речовини**

Оберіть тип викинутої речовини ▼

**Кількість викинутої речовини, т**

Введіть кількість викинутої речовини

**Відсоток працівників забезпечених протигазами**

Введіть відсоток працівників

**Розрахувати**

Рисунок 3.19 – Мапа та форма вводу даних для аналізу викиду НХР

На формі присутні такі поля:

- місце аварії – з бази виводиться перелік хімічно небезпечних об'єктів про які є інформація в базі, користувач повинен один з них;
- тип викинутої речовини – з бази виводиться перелік хімічно небезпечних речовин про які є інформація в базі, користувач повинен одну з них;
- кількість викинутої речовини – вводиться кількість речовини яка була викинута у тонах;
- відсоток працівників забезпечених протигазами – вводиться інформація про забезпеченість працівників підприємства протигазами у відсотках.

Після введення даних та натискання кнопки «Розрахувати», на сторінку додається розділ, в який виводиться мапа з відображеною розрахованою зоною враження, а також таблиці:



– результати оцінки хімічної обстановки – виводиться інформація про розраховані глибину, ширину та площу зони враження, тривалість вражаючої дії та втрати людей які працюють на підприємстві.

– результати оцінки прогнозованого еколого-економічного збитку – виводиться інформація про витрати на відшкодування життя та здоров'я населення, збитки від порушення сільськогосподарських угідь, збитки від втрати деревини та інших лісових ресурсів, та збитки рибного господарства.

Вигляд інформаційного блоку представлено на рисунку 3.20. Загальний вигляд сторінки представлено на рисунку 3.21.



Тип НХР, кількість, тон	Глибина ПЗХЗ, км	Ширина ПЗХЗ, км	Площа ПЗХЗ, км.кв.	Площа МЗХЗ, км.кв.	Тривалість уражаючої дії, хв	Втрати людей, чол
хлорпікрин, 100	4.31	0.9	3.26	7.29	284	Всього: 652, смерт.- 228, серед.- 261, легкі- 163

Таблиця 1 - Результати оцінки хімічної обстановки

Витрати на відшкодування життя та здоров'я населення	Збитки від порушення сільськогосподарських угідь	Збитки від втрати деревини та інших лісових ресурсів	Збитки рибного господарства
13392.94	18.94	7.90	0.00

Таблиця 2 - Результати оцінки прогнозованого еколого-економічного збитку

Рисунок 3.20 – Блок з результатами аналізу



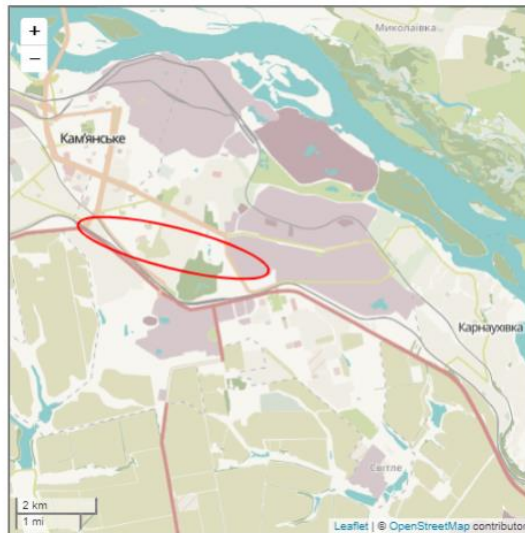
**Місце аварії**

**Тип викинутої речовини**

**Кількість викинутої речовини, т**

**Відсоток працівників забезпечених протигазами**

**Розрахувати**



Тип НХР, кількість, ТОН	Глибина ПЗХЗ, км	Ширина ПЗХЗ, км	Площа ПЗХЗ, км.кв.	Площа МЗХЗ, км.кв.	Тривалість уражаючої дії, хв	Втрати людей, чол
хлорпкрин, 100	4.31	0.9	3.26	7.29	284	Всього: 652, смерт - 228, серед - 261, легкі - 163

Таблиця 1 - Результати оцінки хімічної обстановки

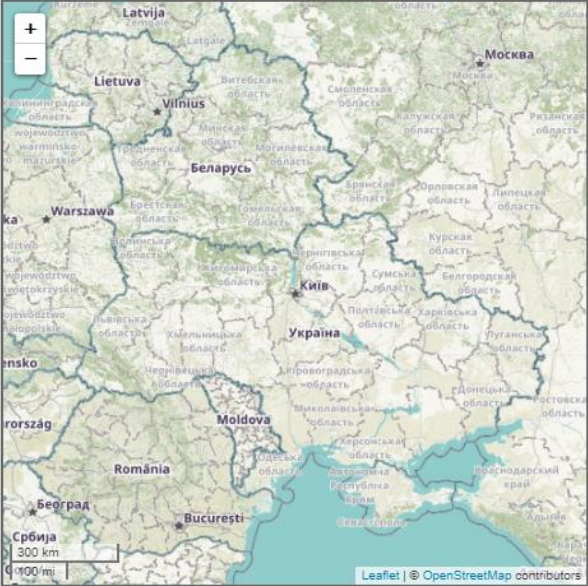
Витрати на відшкодування життя та здоров'я населення	Збитки від порушення сільськогосподарських угідь	Збитки від втрати деревини та інших лісових ресурсів	Збитки рибного господарства
13392.94	18.94	7.90	0.00

Таблиця 2 - Результати оцінки прогнозованого еколого-економічного збитку

Рисунок 3.21 – Загальний вид сторінки аналізу збитків від НХР

Перейшовши на сторінку аналізу збитків від землетрусу(рис. 3.22), користувач бачить мапу та форму для заповнення даних.

Головна
Аварія з викидом НХР
Землетрус
Лісова пожежа



**Варіант вводу координат землетрусу**

--Оберіть варіант вводу--

**Широта**

Введіть широту

**Довгота**

Введіть довготу

**Магнітуду**

Введіть магнітуду

**Глибина гіпоцентру**

Введіть глибину гіпоцентру

Розрахувати

Рисунок 3.22 – Мапа та форма вводу даних для аналізу землетрусу

На формі присутні такі поля:

- варіант вводу координат землетрусу – представляє з себе випадючий список із значеннями: маркер на мапі та координати місця виникнення. Якщо обрано маркер на мапі, то з’являється можливість встановити маркер на мапі, а в поля широта та довгота автоматично будуть занесені дані, які відповідають широті та довготі встановленого маркеру. Якщо обрано координати місця виникнення, то користувачу необхідно ввести координат місця виникнення самостійно;
- широта – широта місця виникнення;
- довгота – довгота місця виникнення;



- магнітуда – значення магнітуди по шкалі Ріхтера;
- глибина гіпоцентру – відстань від поверхні землі на якій виникли поштовхи, км.

Після введення даних та натискання кнопки «Розрахувати», на сторінку додається розділ, в який виводиться мапа з відображеною, розрахованою зоною враження. На мапі рожевим кольором відображається епіцентр виникнення, червоним – зона враження в радіусі 10 км, помаранчевим – зона враження в радіусі 20 км. Також відображаються наступні таблиці:

- наслідки землетрусу в кожній зоні, в залежності від інтенсивності на поверхні – виводиться інформація про інтенсивність землетрусу та опис наслідків, що слід очікувати від землетрусу з такою інтенсивністю. Інформація виводиться по кожній із зазначених зон.

- втрати населення в кожній зоні, в залежності від інтенсивності на поверхні – виводиться інформація про відсоток населення, яке загине в кожній зоні в залежності від того, в будинку якого типу люди будуть знаходитися.

- стійкість систем життєзабезпечення – виводиться інформація про те, який шанс того, що кожна із перелічених систем життєзабезпечення буде працювати в той же день/через 24 години.

Вигляд інформаційного блоку представлено на рисунку 3.23.

Загальний вигляд сторінки представлено на рисунку 3.24.



	Інтенсивність землетрусу	Наслідки землетрусу
В епіцентрі виникнення	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах гір.
В радіусі 10км від епіцентру	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах гір.
В радіусі 20км від епіцентру	7	Значне пошкодження будівель: тріщини в штукатурці і відламування окремих шматків, тонкі тріщини в стінах, пошкодження димових труб. Тріщини в сирих ґрунтах.

Таблиця 1 - Наслідки землетрусу в кожній зоні, в залежності від інтенсивності на поверхні

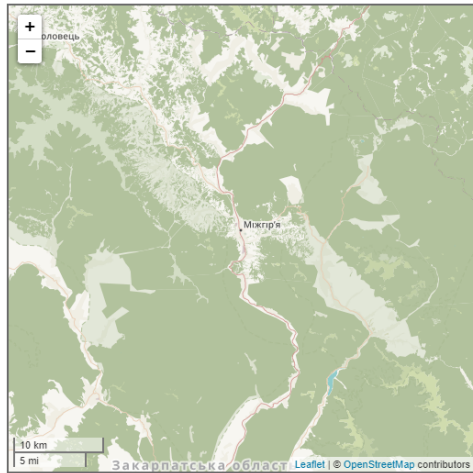
Тип будівлі	Втрати населення в епіцентрі, %	Втрати населення в радіусі 10км, %	Втрати населення в радіусі 20км, %
Дерев'яні	0	0	0
Цегляні малоповерхові (1-2 поверху)	15	15	10
Цегляні багатоповерхові	3	3	0
Цегляні з неповною каркасною стіною	3	3	0

Таблиця 2 - Втрати населення в кожній зоні, в залежності від інтенсивності на поверхні, %

Система	Стійкість систем в епіцентрі, %	Стійкість систем в радіусі 10км, %	Стійкість систем в радіусі 20км, %
водопостачання	48/53	48/53	53/80
електропостачання	60/75	60/75	75/85
газопостачання	77/85	77/85	85/90
теплопостачання	50/77	50/77	77/85
транспорт	68/85	68/85	85/90
каналізація	82/90	82/90	90/100
зв'язок	82/90	82/90	90/100

Таблиця 3 - Стійкість систем життєзабезпечення, %

Рисунок 3.23 – Блок з результатами аналізу



**Варіант вводу координат землетрусу**

Маркер на мапі

**Широта**  
48.52041345737233

**Довгота**  
23.50250244140625

**Магнітуда**  
7

**Глибина гіпоцентру**  
50

**Розрахувати**



	Інтенсивність землетрусу	Наслідки землетрусу
В епіцентрі виникнення	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах пр.
В радіусі 10км від епіцентру	8	Руйнування в будівлях: великі тріщини в стінах, падіння карнизів, димових труб. Зсуви і тріщини шириною до декількох сантиметрів на схилах пр.
В радіусі 20км від епіцентру	7	Значне пошкодження будівель: тріщини в штукатурці і відламування окремих шматків, тонкі тріщини в стінах, пошкодження димових труб. Тріщини в сирих ґрунтах.

Таблиця 1 - Наслідки землетрусу в кожній зоні, в залежності від інтенсивності на поверхні

Тип будівлі	Втрати населення в епіцентрі, %	Втрати населення в радіусі 10км, %	Втрати населення в радіусі 20км, %
Дерев'яні	0	0	0
Цегляні малоповерхові (1-2 поверху)	15	15	10
Цегляні багатоповерхові	3	3	0
Цегляні з неповною каркасною стіною	3	3	0

Таблиця 2 - Втрати населення в кожній зоні, в залежності від інтенсивності на поверхні, %

Система	Стійкість систем в епіцентрі, %	Стійкість систем в радіусі 10км, %	Стійкість систем в радіусі 20км, %
водопостачання	48/53	48/53	53/90
електропостачання	60/75	60/75	75/85
газопостачання	77/85	77/85	85/90
теплопостачання	50/77	50/77	77/85
транспорт	68/85	68/85	85/90
каналізація	82/90	82/90	90/100
зв'язок	82/90	82/90	90/100

Таблиця 3 - Стійкість систем життєзабезпечення, %

Рисунок 3.24 – Загальний вид сторінки аналізу збитків від землетрусу

Перейшовши на сторінку аналізу збитків від лісової пожежі(рис. 3.25), користувач бачить мапу та форму для заповнення даних.

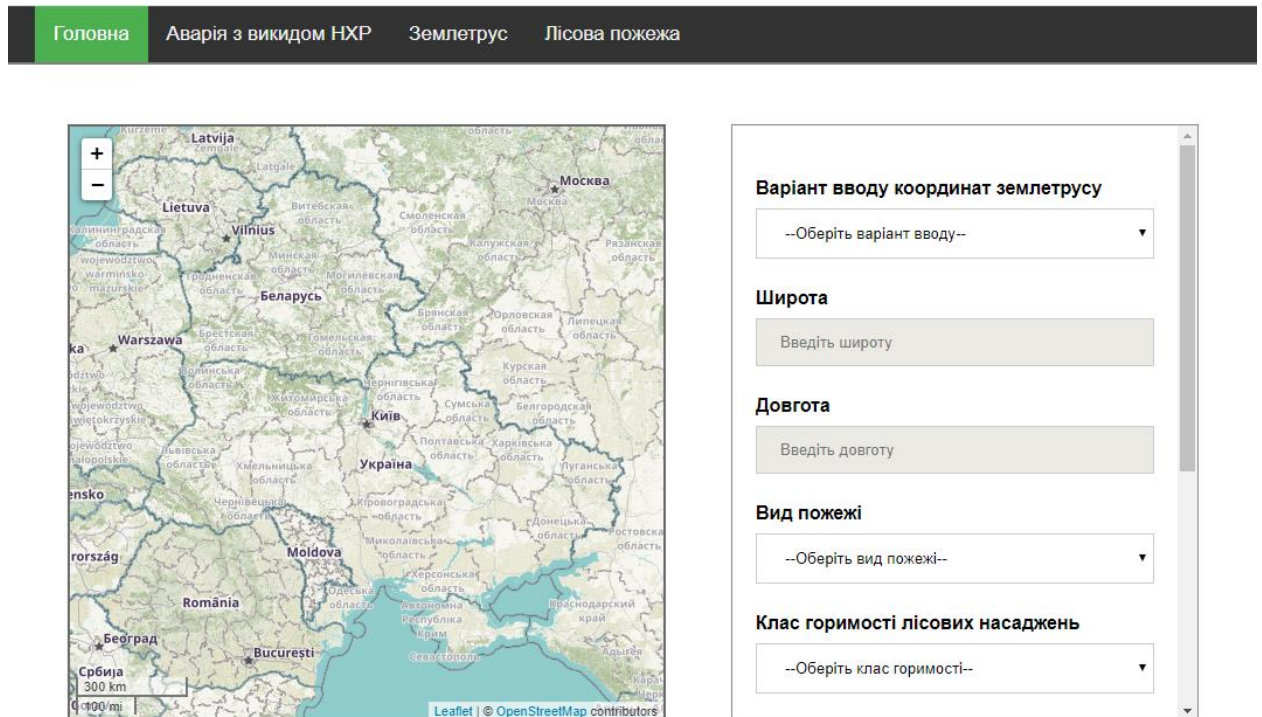


Рисунок 3.25 – Мапа та форма вводу даних для аналізу лісової пожежі

На формі присутні такі поля:

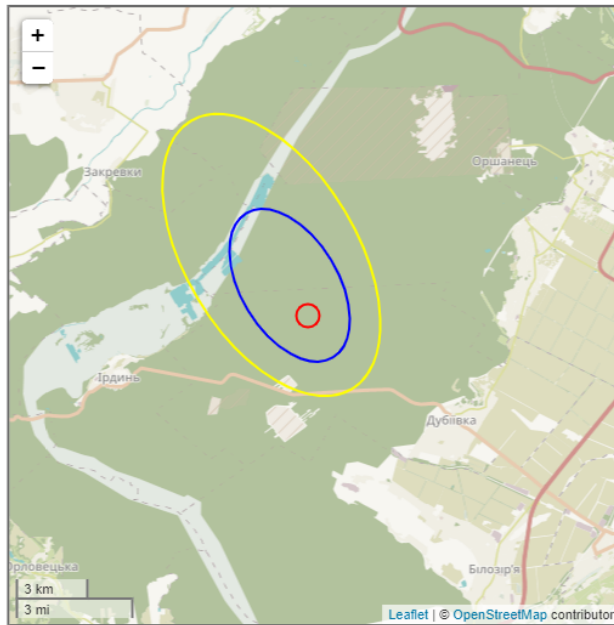
- варіант вводу координат землетрусу – представляє з себе випадючий список із значеннями: маркер на мапі та координати місця виникнення. Якщо обрано маркер на мапі, то з’являється можливість встановити маркер на мапі, а в поля широта та довгота автоматично будуть занесені дані, які відповідають широті та довготі встановленого маркеру. Якщо обрано координати місця виникнення, то користувачу необхідно ввести координат місця виникнення самостійно;
- широта – широта місця виникнення;
- довгота – довгота місця виникнення;

- вид пожежі – верхова або низова;
- клас горимості лісових насаджень – чисті і з домішкою листяних порід хвойні насадження (крім модринових), чисті з домішкою хвойних порід листяні насадження, а також модринові насадження;
- середня висота нагару;
- середній діаметр деревостою;
- початковий периметр загорання.

Після введення даних та натискання кнопки «Розрахувати», на сторінку додається розділ, в який виводиться мапа з відображеною, розрахованою зоною враження. На мапі червоним кольором відображається територія пожежі в момент виявлення, синім – територія пожежі через добу, жовтим – територія пожежі через дві доби. Також відображаються наступні таблиці:

- розповсюдження пожежі в часових проміжках – виводиться інформація про площу лісової пожежі(Га) та периметр лісової пожежі(км);
- загальні відомості про пожежу – виводиться інформація про клас пожежної небезпеки, швидкість фронту, флангів та тилу пожежі, а також відсоток деревини непригідної для реалізації.

Вигляд інформаційного блоку представлено на рисунку 3.26.  
Загальний вигляд сторінки представлено на рисунку 3.27.



Час	Площа лісової пожежі, Га	Периметр лісової пожежі, км
На момент виявлення(червоний)	72	3000
Через добу після виявлення(синій)	0.626	12504
Через дві доби після виявлення(жовтий)	1.938	22008

Таблиця 1 - Розповсюдження пожежі в часових проміжках

Клас пожежної небезпеки	Швидкість фронту пожежі, м/с	Швидкість флангу пожежі, м/с	Швидкість тилу пожежі, м/с	Відсоток деревини непригодної для реалізації
5	120	39	1	Древостой после пожара заметно изреживается; характеризуется сохранением жизнедеятельности значительного количества деревьев верхнего полога и отмиранием подчиненной части древостоя после низовых пожаров средней силы. Відсоток непригодної деревини = 31 - 70

Таблиця 2 - Загальні відомості про пожежу

Рисунок 3.26 – Блок з результатами аналізу



Верховий стійкий

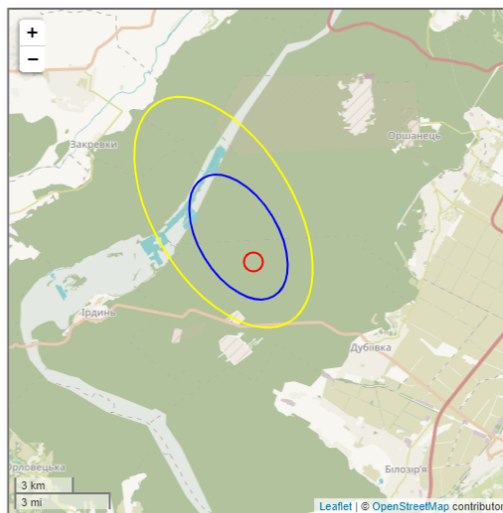
**Клас горимості лісових насаджень**  
Чисті і з домішкою листяних порід хвойні нас

**Середня висота нагару**  
1.0 - 1.5

**Середній діаметр деревостою**  
12

**Початковий периметр зони загорання, м**  
3000

Розрахувати



Час	Площа лісової пожежі, Га	Периметр лісової пожежі, км
На момент виявлення(червоний)	72	3000
Через добу після виявлення(синій)	0.626	12504
Через дві доби після виявлення(жовтий)	1.938	22008

Таблиця 1 - Розповсюдження пожежі в часових проміжках

Клас пожежної небезпеки	Швидкість фронту пожежі, м/с	Швидкість флангу пожежі, м/с	Швидкість типу пожежі, м/с	Відсоток деревини непригодної для реалізації
5	120	39	1	Древостою після пожежа помітно изреживається; характеризується збереженням життєдіяльності значительного количества деревьев верхнего полога и отмиранием подчиненной части древостоя после низовых пожаров средней силы. Відсоток непригодної деревини = 31 - 70

Таблиця 2 - Загальні відомості про пожежу

Авторизувавшись у додатку з правами модератора, відкривається доступ до панелі модератора, яка представляє з себе форму для занесення інформації про хімічно небезпечні об'єкти(рис. 3.28).

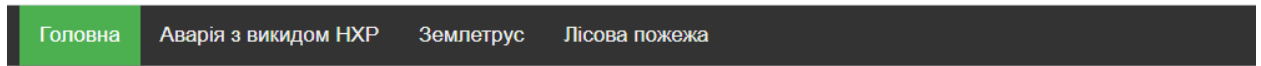
A white rectangular form with a thin grey border. It contains five text input fields, each with a label above it and a placeholder text inside. The labels are: 'Назва об'єкту', 'Широта', 'Довгота', 'Кількість працівників', and 'Місто'. The placeholder texts are: 'Введіть назву об'єкту', 'Введіть широту', 'Введіть довготу', 'Введіть кількість працівників', and 'Введіть місто'. At the bottom of the form is a solid green button with the white text 'Додати'.

Рисунок 3.28 – Панель модератора

На формі присутні наступні поля:

- назва об'єкту;
- широта – широта на якій знаходиться об'єкт;
- довгота – довгота на якій знаходиться об'єкт;
- кількість працівників – інформація про те, яка кількість людей працює на підприємстві, береться модератором з офіційної звітності підприємства;
- місто – місто в якому об'єкт розміщено.

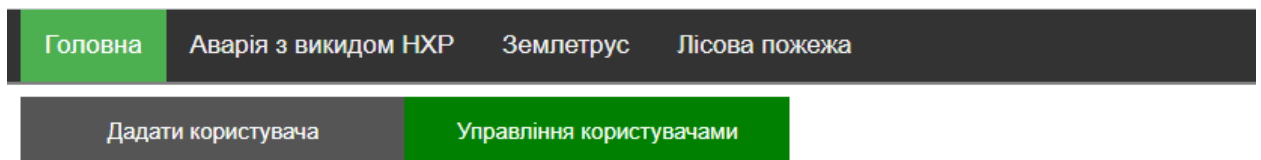


Інформація занесена до форми заноситься в базу знань, після чого використовується для розрахунків збитків від аварії з викидом НХР.

Якщо користувач авторизувався з правами адміністратора, то він отримує доступ до адмін-панелі. Ця сторінка складається з двох вкладок: управління користувачами(рис 2.29) та додати користувача(рис. 2.30).

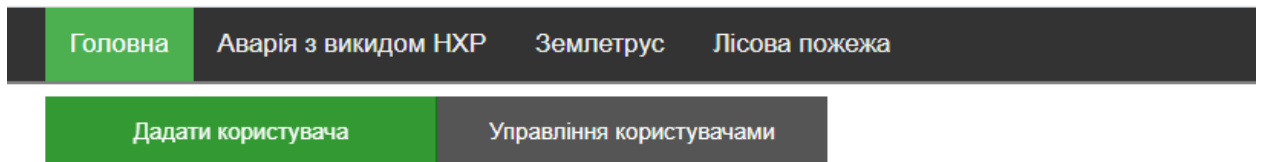
На вкладці управління користувачами відображається таблиця користувачів в системі, їх імена та ролі, а також присутня можливість видалити користувача з системи.

На вкладці додати користувача знаходиться форма реєстрації нового користувача. Вводиться його логін, задається пароль, вручну або за допомогою генератора та задається роль.



id	Логін	Роль	Пароль	Видалити
11	superuser	admin		Видалити
14	alexei	admin		Видалити

Рисунок 2.29 – Вкладка управління користувачами



### Додати користувача

**Логін**

**Пароль**

**Роль**

Рисунок 2.30 – Вкладка додати користувача

## ВИСНОВКИ

В ході виконання дипломної роботи було розроблено web-додаток аналізу збитків від техногенних та природніх катастроф.

Актуальність створення додатку обумовлюється тим, що зараз у світі кожного року виникає все більше катастроф як природного так і техногенного характеру. Тільки на території України знаходиться близько 20 крупних та багато дрібних хімічно-небезпечних підприємства. Також щороку виникає до 3.5 тисяч лісових пожеж, а західна Україна страждає від землетрусів магнітудою в 6-7 балів.

Для отримання додатку, який якомога краще задовольняє потреби користувачів, було проведено аналіз додатків які надають схожі можливості, проведено аналіз існуючих методик аналізу збитків та обрано ті з них які найбільше підходять для поставлених задач.

В ході реалізації додатку, було розроблено модуль роботи за мапою, модуль роботи з API, що надає інформацію про погодні умови, та модулі аналізу збитків від техногенних та природніх катастроф. Результати розробки наведені у вигляді схем, зображень, діаграм та таблиць, розміщених вище в документі. Програмний код розробленого web-додатку наведено в додатку В та його можна переглянути за посиланням на гітхаб репозиторій <https://github.com/alexe4540/diploma>.

Для підвищення якості розробки було проведено моделювання та проектування проекту, розроблено діаграму Ганта, оцінено ризики та розроблено календарний план робіт. Переглянути результати виконаної роботи можна у представленому документі та у додатках А та Б.

Для підвищення якості розробленого додатку було проведено тестування. Усі виявлені дефекти було виправлено.

Область використання – центри екологічної інспекції та потенційно небезпечні промислові підприємства.

Використання додатку дозволить прогнозувати та розраховувати економічні збитки в наслідок техногенної катастрофи, підвищити якість та швидкість усунення наслідків від техногенних катастроф.

Результати роботи були апробовані на науково-практичній конференції ІМА 2020 в Сумському державному університеті.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Основи безпечної життєдіяльності [Електронний ресурс]. – Режим доступу: <http://www.diagram.com.ua/info/obzhd/obzhd60.shtml> – 17.04.2020р. – Класифікація катастроф. ОБЖД.
- 2 Методологія IDEF0 [Електронний ресурс]. – Режим доступу: <https://itteach.ru/bpwin/metodologiya-idef0> – 17.04.2020р. – Методологія IDEF0.
- 3 Ткачук А.І. Цивільний захист / А.І. Ткачук, О.В. Пуляк – Кропивницький, 2017 – 144с.
- 4 UML Use Case Diagrams [Електронний ресурс]. – Режим доступу: <https://www.uml-diagrams.org/use-case-diagrams.html> – 18.04.2020р. – Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors).
- 5 Responsibility Assignment Matrix [Електронний ресурс]. – Режим доступу: <http://acqnotes.com/acqnote/careerfields/responsibility-assignment-matrix1> – 19.04.2020р. – Responsibility Assignment Matrix (RAM) - AcqNotes.
- 6 Ю. А. Опанасюк, Еколого-економічна оцінка техногенних катастроф [Текст]: дисертація на здобуття наукового ступеня канд. економічних наук / — Суми : СумДУ, 2015. — 242 с.
- 7 Методика розрахунку параметрів лісових пожеж [Електронний ресурс]. – Режим доступу: <https://cyberleninka.ru/article/n/metodika-rascheta-parametrov-lesnyh-pozharov-kak-dinamicheskikh-protsessov-na-poverhnosti-zemli-s-ispolzovaniem-dannyh-kosmicheskogo/viewer> – 18.04.2020р. – Методика расчета параметров лесных пожаров как динамических процессов на поверхности земли.
- 8 Microsoft Developer Network [Електронний ресурс] – Режим доступу до ресурсу: <https://msdn.microsoft.com>.
- 9 Stack Overflow [Електронний ресурс] – Режим доступу до ресурсу: <http://stackoverflow.com/>.

- 10 Habrahabr [Электронный ресурс] – Режим доступа до ресурсу: <https://habrahabr.ru/>.
- 11 Github [Электронный ресурс] – Режим доступа до ресурсу: <https://github.com/>.
- 12 Офіційний сайт «Портал знань» [Электронный ресурс] – Режим доступа : <http://www.znannya.org/>.
- 13 Шевчук А., Касьянов А. Patterns via C#. Приемы объектно-ориентированного проектирования: ITVDN / А. Шевчук, Д. Охрименко, 2015 – 288с.
- 14 Орлов, С.А. Технология разработки программ / С. А. Орлов – СПб.: Питер, 2004.
- 15 Федорчук, Д. А. Разработка WEB приложений на PHP и MySQL / Д.А. Федорчук - СПб. : Корона-принт, 2003.
- 16 Овицкий, Н.Ф. PHP как инструмент для работы с MySQL базой данных / Л.З Овицкий - М. : Вильямс, 2008.
- 17 Прохоров, В.В. Разработка WEB приложений с нуля / В.В. Прохоров - Мн.: БГУИР, 2009.
- 18 Leaflet [Электронный ресурс] – Режим доступа до ресурсу: <https://leafletjs.com/>.
- 19 Dark Sky API [Электронный ресурс] – Режим доступа до ресурсу: <https://darksky.net/dev>
- 20 Part 1. How to Use the Dark Sky API [Электронный ресурс] – Режим доступа до ресурсу: <https://github.com/initialstate/wunderground-sensehat/wiki/Part-1.-How-to-Use-the-Dark-Sky-API>
- 21 Part 3. Hyper Local Weather Dashboard [Электронный ресурс] – Режим доступа до ресурсу: <https://github.com/initialstate/wunderground-sensehat/wiki/Part-3.-Hyper-Local-Weather-Dashboard>
- 22 Leaflet: Make a web map! [Электронный ресурс] – Режим доступа до ресурсу: <https://maptimeboston.github.io/leaflet-intro/>

# **ДОДАТОК А**

## **ТЕХНІЧНЕ ЗАВДАННЯ**

**на розробку «Web-додаток аналізу збитків від техногенних та природніх катастроф»**

**Суми 2020**

# **1 Призначення й мета створення web-додатку**

## **1.1 Призначення web-додатку**

Даний web-додаток призначений для виконання аналіз збитків від техногенних та природніх катастроф.

## **1.2 Мета створення web-додатку**

Забезпечення оперативного аналізу збитків із мінімальним впливом людини на процес.

## **1.3 Цільова аудиторія**

До цільової аудиторії web-додатку можна віднести наступні групи:

1. МНС України.
2. Хімічні заводи.
3. Організації які слідкують за станом довкілля.
4. Інші зацікавлені сторони.



## **2 Вимоги до web-додатку**

### **2.1 Вимоги до web-додатку в цілому**

#### **2.1.1 Вимоги до структури й функціонування web-додатку**

ІС має бути виконана у вигляді web-додатку з інтуїтивно зрозумілим інтерфейсом. Для зменшення похибки та часу виконання розрахунків, вплив людини на систему повинен бути мінімальним. Вихідні дані повинні подаватися у зручній та зрозумілій формі.

#### **2.1.2 Вимоги до персоналу**

Для роботи з системою під обліковим записом звичайного користувача немає необхідності мати спеціальні навички та знання, достатньо базових навичок роботи з персональним комп'ютером (ПК).

Для адміністрування web-додатку, а саме: оновлення даних в інформаційній базі, внесення до неї інформації про хімічні заводи та інші аварійно небезпечні об'єкти і виконання інших дій по адмініструванню системи, людина повинна мати знання в галузі основ цивільного права та має бути знайома з системою адміністрування web-додатку.

#### **2.1.3 Вимоги до збереженні інформації**

Вся основна інформація повинна зберігатися в інформаційній базі.

#### **2.1.4 Вимоги до розмежування доступу**

Web-додаток є публічним, тому доступ до його використання є загальнодоступним, окрім функцій, які відповідають за адміністрування та налаштування web-додатку.

Користувачів web-додатку можна розділити на 3 групи:

1. Відвідувачі.

2. Користувачі з правом читання-запису.

3. Головний адміністратор.

**Відвідувачі** мають доступ тільки до тієї частини додатку, яка є загальнодоступною, а саме задання вхідних даних та перегляд результатів аналізу збитків.

**Користувачі з правом читання-запису** мають можливість як виконувати розрахунки, так і додавати та редагувати інформацію про аварійно небезпечні об'єкти.

**Адміністратор** має повний доступ до налаштування web-додатку, а саме:

- додавання користувачів з правами читання-запису;
- редагування, видалення та додавання інформації в інформаційній бази.

## **2.2 Вимоги до функцій, виконуваних сайтом**

### **2.2.1 Основні вимоги**

#### **2.2.1.1 Структура сайту**

Сайт повинен складатися з наступних розділів:

Головна сторінка – виводить основну інформацію про web-додаток, її призначення та посилання на інші розділи додатку.

Аналіз збитків від техногенної катастрофи – виконуються всі необхідні розрахунки для аналізу збитків від техногенної катастрофи та виводиться результат аналізу.

Аналіз збитків від природної катастрофи – виконуються всі необхідні розрахунки для аналізу збитків від природної катастрофи та виводиться результат аналізу.

Панель користувача з правами читання-запису – надає функціонал для додавання потенційних об’єктів аварій.

Панель адміністратора – надає функціонал для повного доступу до інформаційної бази, запису, редагування та видалення даних, створення нових користувачів.

### **2.2.1.2 Навігація**

Інтерфейс користувача повинен бути простим та зрозумілим, доступ до необхідного функціоналу повинен отримуватися за декілька кліків, на сторінках повинні бути присутні «хлібні крихти» для простішої навігації.

ІС має надавати користувачеві доступ до всіх доступних йому ресурсів. Всі посилання повинні мати зрозумілі підписи які забезпечують однозначне розуміння змісту сторінок на які потрапить користувач.

### **2.2.1.3 Наповнення сайту (контент)**

Всі дані, необхідні для розрахунку, зберігаються в інформаційній базі на сервері або зчитуються з сторонніх АРІ, наприклад для отримання погодних умов під час виникнення катастрофи. Дані з бази зчитуються в режимі реального часу в залежності від введених користувачем даних.

Наповнення додатку інформацією необхідною для розрахунків відбувається двома способами:

- З панелі користувача з правами читання-запису.
- З адмін-панелі, яка забезпечує повний контроль над інформацією в інформаційній базі.

### **2.2.1.4 Система навігації (карта сайту)**

Взаємозв’язок між розділами й підрозділами сайту (карта сайту) представлено на рисунку А.1.



Рисунок А.1– Карта сайту

### 2.2.2 Вимоги до функціональних можливостей

ІС повинна надавати користувачам функціонал, для виконання аналізу збитків, інформація про погодні умови на момент катастрофи повинні братися з АРІ погоди, зона враження повинна відображатися на мапі або зображенні мапи, результати аналізу повинні виводиться у вигляді зведеної таблиці даних. Повинен бути функціонал для внесення потенційних місць аварій до бази знань, а також її редагування.

#### 2.2.2.1 Функціональні можливості розділів

Головна сторінка додатку повинна містити наступні розділ:

- Шапка сайту:
  - Логотип;
  - «Хлібні крихти» для навігації;
- Загальна інформація про web-додаток;

- Блок опису модулю аналіз збитків від техногенних катастроф з можливістю переходу до розрахунку;
- Блок опису модулю аналіз збитків від природних катастроф з можливістю переходу до розрахунку;
- Підвал сайту:
  - Контактні дані;
  - Перехід до формі авторизації адміністратора або користувача з правами читання-запису.

### **2.2.2.2 Загальні вимоги**

Дизайн головної сторінки додатку повинен бути простим та лаконічним, не повинно бути відволікаючих елементів, кольори спокійні та не викликають втому людського ока.

Дизайн сторінок розрахунку схожий, по центру екрану форма для вхідних даних, після виконання додаються необхідні блоки, такі як зведена таблиця з даними або мапа, з відображеними зонами враження на ній.

Розташування елементів на головній сторінці сайту та сторінках аналізу схематично показано на рис. А.2-А.3.

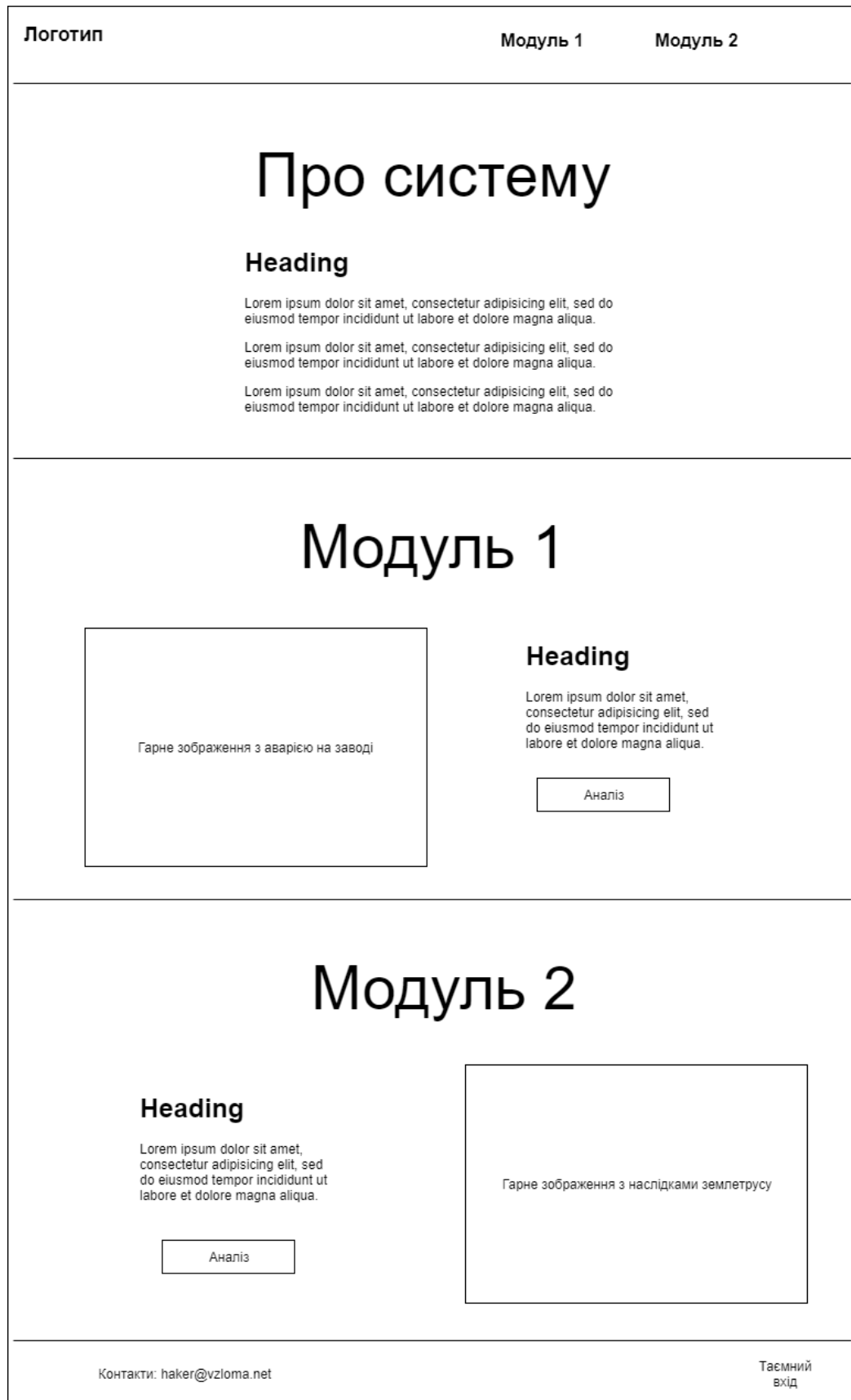


Рисунок А.2- Головна сторінка

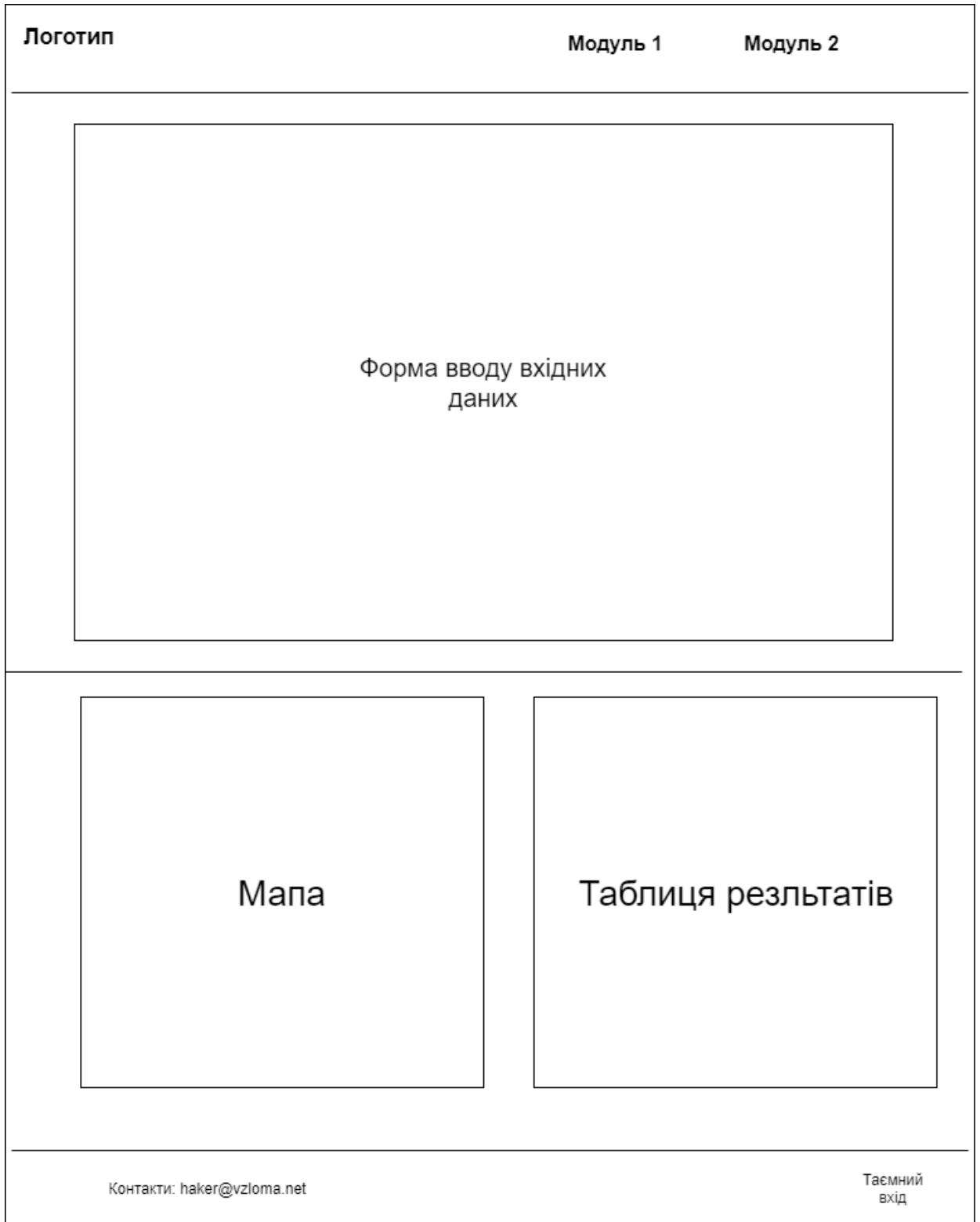


Рисунок А.3 – Схематичне зображення сторінки аналізу збитків

### **2.2.2.3 Типові навігаційні й інформаційні елементи**

Шапка сайту;

Інформаційний блок;

Підвал сайту.

### **2.2.2.4 Шапка сайту**

Шапка сайту повинна містити логотип та «хлібні крихти». Логотип є посиланням на головну сторінку, «хлібні крихти» ведуть на сторінки з розрахунками.

### **2.2.2.5 Інформаційний блок**

Основний блок - відображає інформацію типу, заголовок, опис та зображення. Інформація, яка відображається, стосується певного розділу, в собі має посилання на модуль розрахунку який стосується даного розділу.

### **2.2.2.6 Підвал сайту**

Містить контактні дані та посилання для входу користувача з правами читання-запису та адміністратора.

## **2.3 Вимоги до видів забезпечення**

### **2.3.1 Вимоги до інформаційного забезпечення**

Реалізація додатку відбувається з використанням:

- Apache 2.4;
- PHP 7.2;
- MySQL 8.0;
- CSS 3.0;
- HTML 5.0;



- JavaScript;
- OpenStreetMap;
- Dark Sky API.

### 2.3.2 Вимоги до лінгвістичного забезпечення

Web-додаток повинен бути виконаний українською мовою.

### 2.3.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- наявний доступ до мережі Інтернет;
- Операційна система: Microsoft Windows 7(і вище);
- Веб-браузер з підтримкою ECMAScript 6.

### 2.3.4 Вимоги до апаратного забезпечення

Апаратне забезпечення серверної частини повинне задовольняти наступним вимогам:

- Веб-сервер Apache;
- Не менш 500 МБ вільного місця на диску.

Апаратне забезпечення клієнтської частини повинне забезпечувати підтримку програмного забезпечення клієнтської частини, зазначеного в п.

2.3.3

### 3 Склад і зміст робіт

Докладний опис етапів роботи розробки web-додатку наведено на рисунку А.4 у вигляді списку робіт для побудови діаграми Ганта.

	Режим задачі	Название задачи	Длительнс	Начало	Окончани	Предшественн
1		<b>Формування технічного завдання</b>	<b>8 дней</b>	<b>Пн 06.04.20</b>	<b>Ср 15.04.20</b>	
2		Визначення призначення й мети ІС	2 дней	Пн 06.04.20	Вт 07.04.20	
3		Визначення вимог до ІС	2 дней	Ср 08.04.20	Чт 09.04.20	2
4		Визначення складу і змісту робіт	2 дней	Пт 10.04.20	Пн 13.04.20	3
5		Визначення вимог до введення ІС в експлуатацію	1 день	Вт 14.04.20	Вт 14.04.20	4
6		Визначення вимог до документації	1 день	Ср 15.04.20	Ср 15.04.20	5
7		<b>Розробка програмного продукту</b>	<b>20 дней</b>	<b>Чт 16.04.20</b>	<b>Ср 13.05.20</b>	
8		Розробка та наповнення інформаційної бази	5 дней	Чт 16.04.20	Ср 22.04.20	6
9		Розробка модулю для зв'язку з Dark Sky API	2 дней	Чт 23.04.20	Пт 24.04.20	8
10		Розробка модулю для роботи з мапою за допомогою OpenStreetMap API	2 дней	Пн 27.04.20	Вт 28.04.20	9
11		Розробка модулю основних розрахунків	5 дней	Ср 29.04.20	Вт 05.05.20	10
12		Розробка модулю адміністрування	4 дней	Ср 06.05.20	Пн 11.05.20	11
13		Розробка користувальницького інтерфейсу	2 дней	Вт 12.05.20	Ср 13.05.20	12
14		<b>Тестування</b>	<b>19 дней</b>	<b>Пн 27.04.20</b>	<b>Чт 21.05.20</b>	
15		Модульне тестування	14 дней	Пн 27.04.20	Чт 14.05.20	9
16		Інтеграційне тестування	14 дней	Ср 29.04.20	Пн 18.05.20	10
17		Системне тестування	3 дней	Вт 19.05.20	Чт 21.05.20	13;15;16
18		<b>Здача програмного продукту</b>	<b>6 дней</b>	<b>Пт 22.05.20</b>	<b>Пт 29.05.20</b>	
19		Розробка документації	5 дней	Пт 22.05.20	Чт 28.05.20	17
20		Введення в експлуатацію	1 день	Пт 29.05.20	Пт 29.05.20	19

Рисунок А.4 – Список робіт для побудови діаграма Ганта

### 4 Вимоги до складу й змісту робіт із введення сайту в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного сайту вимогам сьогодення ТЗ і можливість його

ефективної роботи, в організації Замовника повинен бути проведений певний комплекс заходів.

Для переносу сайту на хостинг необхідно, щоб параметри хостингу відповідали вимогам, зазначеним у ТЗ. На хостинг переноситься програма (сайт), зверстаний шаблон дизайну й структура й наповнення бази даних з подальшою їх доробкою.

## ДОДАТОК Б

### ПЛАНУВАННЯ РОБІТ

#### 1 Ідентифікація ідеї проекту

Метою кваліфікаційної роботи бакалавра є розроблення web-додатку, що буде використовуватися зацікавленими особами для аналізу збитків від техногенних або природніх катастроф.

Дипломний проект призначений для того, щоб зменшити вплив людського фактору на процес аналізу та підвищити швидкість розрахунків у порівнянні з розрахунками вручну.

Web-додаток повинен бути реалізований у повній мірі та доступний в мережі Інтернет. Web-додатку повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

#### 2 Деталізація мети методом SMART

Конкретна (Specific). Розробити WEB-додаток, для аналізу збитків від катастроф техногенного або природнього характеру.

Вимірювана (Measurable). Результатом виконання проекту, має бути Web-додаток, отримати доступ до якого, можна ввівши адресу сайту до адресного рядку, в свою чергу, Web-додаток мусить бути розміщений на хостингу, пов'язані з ним бази даних(БД), форми та модулі мають бути вірно налаштовані та поєднані.

Досяжна (Achievable). Проект потребує розробника із знанням HTML, CSS, мови програмування JavaScript, PHP, MySQL та об'єктно-орієнтованого програмування і програмного забезпечення (ПЗ), яке використовується для розроблення даного додатку.

Реалістична (Relevant). Мету реально досягнути, так як розробка WEB-додатку за допомогою сучасних Web-технологій, не вимагає надзвичайно складних дій або велику кількість ресурсів.

Обмежена у часі (Time—framed). Обмеженість в часі зумовлена рішенням замовника, щоб отримати програмне забезпечення як можна швидше.

### **3 Описання фази розробки ІТ—проекту**

#### **3.1 Планування змісту структури робіт ІТ—проекту (WBS)**

Структурна декомпозиція робіт (work breakdown structure, WBS) - це ієрархічна структура робіт, побудована з метою логічного розподілу усіх робіт з виконання проекту і подана у графічному вигляді. Це сукупність декількох рівнів, кожний з яких формується в результаті розподілу роботи попереднього рівня на її складові. Елементом найнижчого рівня є група робіт, або так званий робочий пакет (work package).

Виконаємо побудову WBS структури, у якій зазначимо всі виконувані роботи в залежності від головних етапів:

1. Формування технічного завдання – перший етап роботи над проектом, під час якого обумовлюються основні вимоги до майбутнього web-додатку. Технічне завдання включає в себе: визначення предметної області, призначення продукту, визначення засобів розробки, визначення цільової аудиторії, розробка макету інтерфейсу та інше.

2. Розробка програмного продукту – написання відповідних модулів, що забезпечують функціонування програмного продукту:

- Розробка та наповнення інформаційної бази.
- Розробка модулю для зв'язку з Dark Sky API.
- Розробка модулю для роботи з мапою за допомогою OpenStreetMap API.
- Розробка модулю основних розрахунків.
- Розробка модулю адміністрування.

- Розробка користувальницького інтерфейсу.

3. Тестування – перевірка роботи програмного продукту, виявлення помилок:

- Модульне тестування.
- Інтеграційне тестування.
- Системне тестування.

4. Здача програмного продукту:

- Розробка документації.
- Введення в експлуатацію.

WBS-структура для даного проекту представлена на рис. Б.1



Рисунок Б.1 - WBS-структура web-додатку аналізу збитків від природних або техногенних катастроф

### 3.2 Планування структури організації, для впровадження готового проекту (OBS)

Наступним кроком розробки структури проекту є визначення організаційної структури (OBS) проекту.

Організаційна структура проекту (OBS) – це графічне відображення учасників проекту, які відповідають та які відносяться до певної частини робіт.

Визначається з елементарних робіт(робіт найнижчого рівня) кожної з гілок WBS. На верхньому рівні знаходиться керівник та команда управління проектом, на наступних – відповідальні особи.

OBS-структура для даного проекту представлена на рисунку Б.2

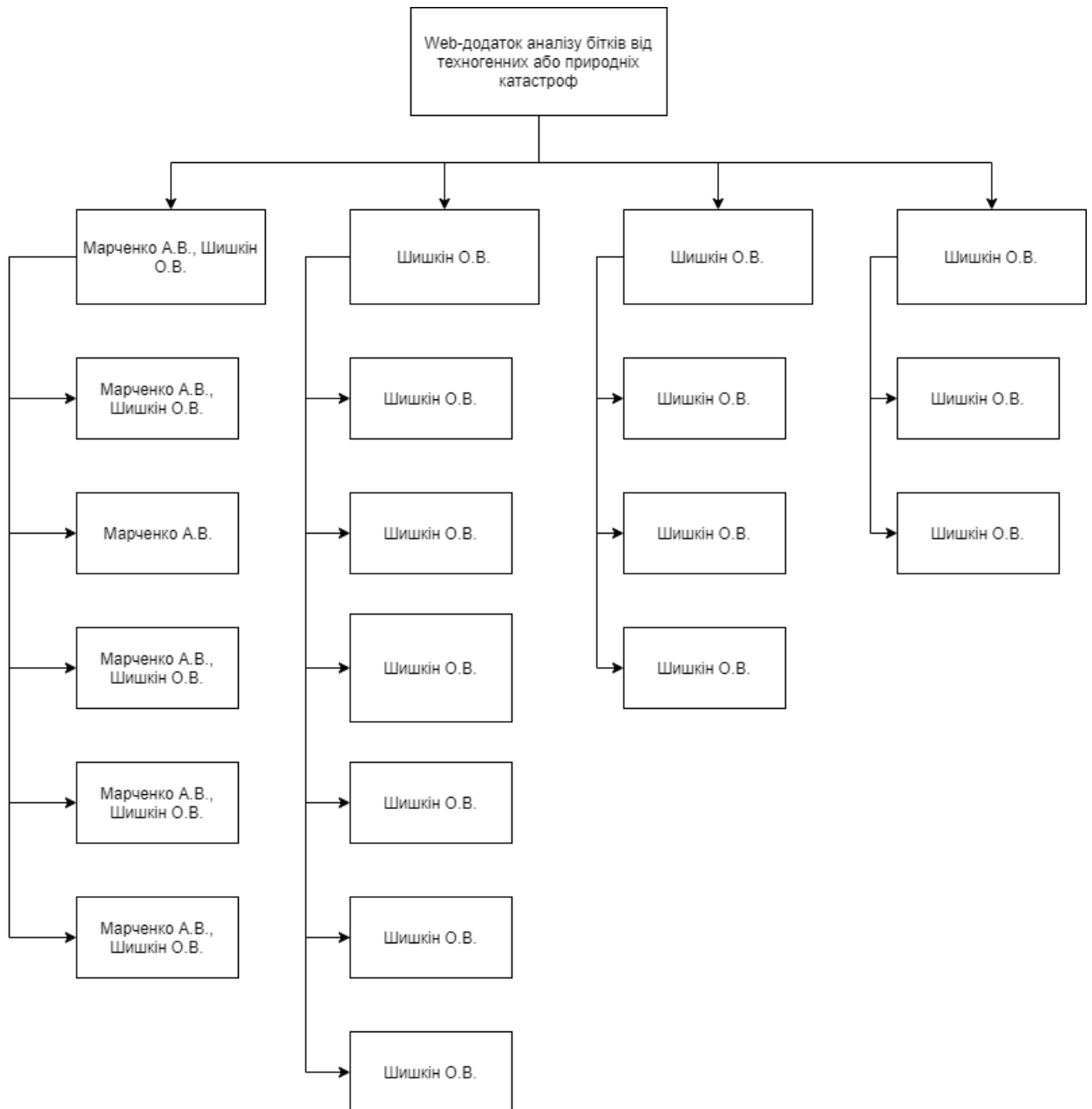


Рисунок Б.2 - Організаційна структура виконавців



### 3.3 Побудова матриці відповідальності (виконавців пакетів робіт)

Матриця відповідальності (Responsibility Assignment Matrix) забезпечує відповідальних за виконання елементарних робіт. Будується на основі WBS та OBS [5]. Для кожного з виконавців була визначена його роль:

Для кожної людини та роботи було виділено наступні ступені відповідальності:

Responsible – відповідальний.

Assists – включений в роботу.

Informed – проінформований.

Матриця відповідальності представлена в табл. Б.1

Таблиця Б.1 – Матриця відповідальності

WBS\OBS	Шишкін О.В.	Марченко А.В	Рецензент
Визначення призначення й мети web-додатку	R	A	
Визначення вимог до web-додатку	I	R	
Визначення складу і змісту робіт	R	A	
Визначення вимог до введення web-додатку в експлуатацію	R	A	
Визначення вимог до документації	R	A	
Розробка та наповнення інформаційної бази	R	I	A
Розробка модулю для зв'язку з Dark Sky API	R	I	
Розробка модулю для роботи з мапою за допомогою OpenStreetMap API	R	I	
Розробка модулю основних розрахунків	R	I	A

Продовження таблиці Б.1

WBS\OBS	Шишкін О.В.	Марченко А.В	Рецензент
Розробка модулю адміністрування	R	I	
Розробка користувальницького інтерфейсу	R	I	
Модульне тестування	R	I	
Інтеграційне тестування	R	I	
Системне тестування	R	I	
Розробка документації	R	I	
Введення в експлуатацію	R	I	

#### **4 Побудова календарного графіку виконання ІТ—проекту (включаючи побудову часткових мережевих моделей у вигляді діаграм Ганта)**

Діаграма Ганта – лінійна діаграма, яка показує задачі проекту, що представляються часовими відрізками, які характеризуються датами початку та закінчення робіт. Діаграма дозволяє відслідковувати відсоток робіт, виконаних по кожному завданню. Цей графік представлено за допомогою програмного засобу MS Project.

Графік виконання дипломного проекту представлено у вигляді Діаграми Ганта на рисунку Б.3. Список робіт для побудови діаграми Ганта представлено на рисунку Б.4

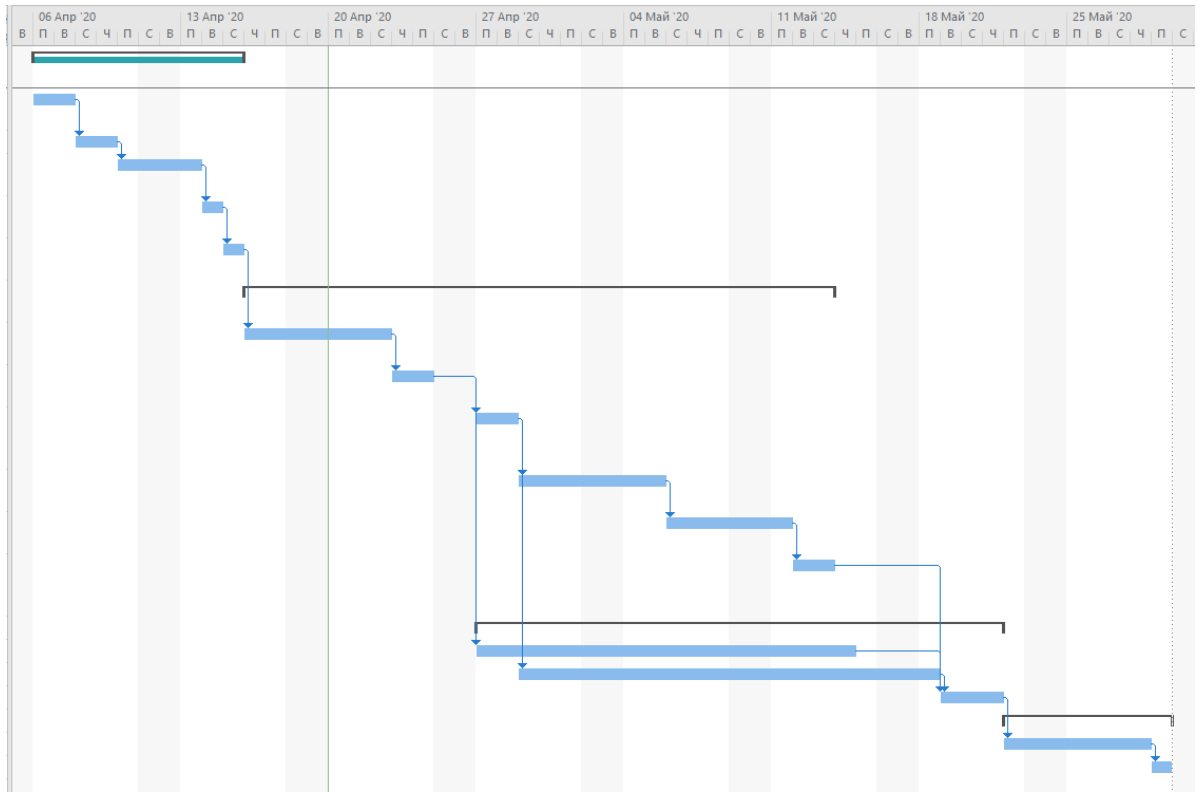


Рисунок Б.3 – Діаграма Ганта

	Режим задачі	Назва задачі	Длительность	Начало	Окончание	Предшественн
1		<b>Формування технічного завдання</b>	8 днів	Пн 06.04.20	Ср 15.04.20	
2		Визначення призначення й мети ІС	2 днів	Пн 06.04.20	Вт 07.04.20	
3		Визначення вимог до ІС	2 днів	Ср 08.04.20	Чт 09.04.20	2
4		Визначення складу і змісту робіт	2 днів	Пт 10.04.20	Пн 13.04.20	3
5		Визначення вимог до введення ІС в експлуатацію	1 день	Вт 14.04.20	Вт 14.04.20	4
6		Визначення вимог до документації	1 день	Ср 15.04.20	Ср 15.04.20	5
7		<b>Розробка програмного продукту</b>	20 днів	Чт 16.04.20	Ср 13.05.20	
8		Розробка та наповнення інформаційної бази	5 днів	Чт 16.04.20	Ср 22.04.20	6
9		Розробка модулю для зв'язку з Dark Sky API	2 днів	Чт 23.04.20	Пт 24.04.20	8
10		Розробка модулю для роботи з мапою за допомогою OpenStreetMap API	2 днів	Пн 27.04.20	Вт 28.04.20	9
11		Розробка модулю основних розрахунків	5 днів	Ср 29.04.20	Вт 05.05.20	10
12		Розробка модулю адміністрування	4 днів	Ср 06.05.20	Пн 11.05.20	11
13		Розробка користувальницького інтерфейсу	2 днів	Вт 12.05.20	Ср 13.05.20	12
14		<b>Тестування</b>	19 днів	Пн 27.04.20	Чт 21.05.20	
15		Модульне тестування	14 днів	Пн 27.04.20	Чт 14.05.20	9
16		Інтеграційне тестування	14 днів	Ср 29.04.20	Пн 18.05.20	10
17		Системне тестування	3 днів	Вт 19.05.20	Чт 21.05.20	13;15;16
18		<b>Здача програмного продукту</b>	6 днів	Пт 22.05.20	Пт 29.05.20	
19		Розробка документації	5 днів	Пт 22.05.20	Чт 28.05.20	17
20		Введення в експлуатацію	1 день	Пт 29.05.20	Пт 29.05.20	19

Рисунок Б.4 – Список робіт для побудови діаграма Ганта

## 5 Ідентифікація ризиків

Ризик – це імовірна подія, яка у випадку своєї появи негативно або позитивно вплине на проект.

Процес управління ризиками включає наступні етапи:

1. Ідентифікація.
2. Процес оцінювання ризиків, який включає в себе якісний, кількісний аналіз.
3. Заходи реагування на ризики.
4. Моніторинг заходів і ризиків.

## Ризики проекту представлені у таблиці Б.2

Таблиця Б.2 – Ризики проекту

№	Ризик	Вплив	Ймовірність	РР	Усунення
1	Неповне ТЗ	4	2	С	Тісний контакт с замовником, що дозволить при виявленні неповноти ТЗ доповнити його
2	Недостатня кваліфікація кадрів	2	2	Н	Відвідування кадрами компанії курсів підвищення кваліфікації
3	Помилки під час планування	3	2	С	Піл час планування адекватно оцінювати можливості
4	Незаплановані завдання які потрібно виконати на пізніх етапах виконання проекту	3	1	Н	Оцінити вплив завдання на хід роботи над проектом, та в разі необхідності домовитися з замовником про збільшення часу на виконання проекту
5	Складні у виконанні або неможливі для виконання завдання	5	2	С	Оновити кадри або домовитися з замовником відмінити завдання

## 6.1 Матриця ризиків

Ризики представити за допомогою RBM матриці (Risk Breakdown Matrix).

Класифікація ризиків за імовірністю виникнення:

- слабоймовірний;
- малоймовірний;
- імовірні;
- досить імовірні;
- майже імовірні.

Класифікація ризиків за величиною втрат:

- мінімальна;
- низька;
- середня;
- висока;
- максимальна.

Виконаємо класифікацію ризиків даного проекту. Для цього складемо таблицю Б.3.

Таблиця Б.3 – Probability / Impact Matrix

5					
4					
3					
2		2	3	1	5
1			4		
Probability / Impact	1	2	3	4	5

## ДОДАТОК В

### АПРОБАЦІЯ

*СЕКЦІЯ 2: Інформаційні  
технології проектування*

ІМА :: 2020

#### **Інформаційна система для аналізу збитків від техногенних або природних катастроф**

Шишкін О.В., студент; Марченко А.В., доцент  
Сумський державний університет, м. Суми, Україна

Ми живемо в час цифрових технологій, коли кожен може знайти будь-яку інформацію, виконавши декілька команд у своєму комп'ютері або смартфоні, обчислювальна потужність яких у десятки разів більше за людську. Саме зараз настав час часткової або повної автоматизації багатьох виробничих і невиробничих процесів. Вже існує множина інформаційних систем (ІС), більшість із яких є Web-додатками. Вони користуються великим попитом через цілодобовий доступ з будь-якого девайсу, що має підключення до мережі інтернет, та відсутність потреби в спеціальних навичках для їх використання.

На сьогоднішній день, в усьому світі, одними з глобальних проблем є проблеми техногенних та природних катастроф. Процес аналізу збитків вимагає послідовності певної кількості розрахунків. Для спрощення процесу розрахунків збитків та зменшення витрат часу на їх виконання, доцільно буде використовувати ІС, яка частково або повністю автоматизує даний процес.

Основною метою проекту є розробка ІС для автоматизації процесу аналізу збитків катастроф різного характеру. Для досягнення мети необхідно розв'язати такі питання: ретельний аналіз існуючих методів аналізу збитків від техногенних або природних катастроф, огляд продуктів, що вирішують аналогічні задачі, проектування та реалізація Web-орієнтованої інформаційної системи. При виконанні проекту були реалізовані такі етапи аналізу збитків: оцінка становища при аварії; графічне відображення зони враження на live-мапі (побудова хмари розповсюдження на карті); розрахунок матеріальних збитків в результаті катастрофи та термінів відновлення ресурсів.

Використання розробленого додатку дозволить виконувати прогнозування потенційних та розрахунки реальних економічних збитків, підвищить точність розрахунків за рахунок того, що деякі дані будуть братися з сторонніх API, а це, в свою чергу, зменшить кількість впливу людського фактору. Додатково дану систему доцільно використовувати при аналізі місцевості при плануванні розташування нового потенційно небезпечного підприємства.

# ДОДАТОК Г

## ЛІСТИНГ ПРОГРАМНОГО КОДУ

### Index.php

```
<? session_start() ?>

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="css/main.css">
  <link rel="stylesheet" href="css/mainpage.css">
</head>

<body>
  <?php include ('./php/module/header.html'); ?>

  <div class="Site-content">
    <section>
      <div class="">
        <div class="img-bg">
          <div class="box">
            <h1>Аналіз збитків від природніх <br>
            та техногенних катастроф</h1>
          </div>
        </div>
      </div>
    </section>
    <section>
      <div class="container">
        <div class="decription-block">
          
          <div class="decription">
            <h1>Аварія з викидом НХР</h1>
            <p> – це подія техногенного характеру, що
            сталася на хімічно небезпечному об'єкті
            внаслідок виробничих, конструктивних,
технологічних чи експлуатаційних причин або від
            випадкових зовнішніх впливів, що призвела до
пошкодження технологічного обладнання,
            пристроїв, споруд, транспортних засобів з виливом
(викидом) НХР в атмосферу і реально
            загрожує життю, здоров'ю людей.</p>
            <a href="/pages/nhrCalculator.php">Перейти до
розрахунку</a>
          </div>
        </div>
      </div>
    </section>
  </div>
</body>
```



```

</section>
<section>
  <div class="container">
    <div class="decription-block">
      
      <div class="decription">
        <h1>Землетрус</h1>
        <p> - це сейсмічні явища, які виникають унаслідок
глибоких шарах Землі
на великі від-
розрахунку</a>
        або внаслідок вулканічних і обвальних явищ, коли
        стані передаються потужні хвилі</p>
        <a href="/pages/earthquakeCalculator.php">Перейти до
      </div>
    </div>
  </div>
</section>
<section>
  <div class="container">
    <div class="decription-block">
      
      <div class="decription">
        <h1>Лісова пожежа</h1>
        <p>- стихійне поширення вогню території лісового
фонду. Пожежа вважається
ділянки чагарників, захарашені вирубки,
на ділянках тимчасового
лісових господарств</p>
        лісовою, коли горять не тільки насадження, але й
        сухий травостій на прогалинах і вирубках, стерня
        сільськогосподарського користування на території
        <a href="fireCalculator.php">Перейти до
розрахунку</a>
      </div>
    </div>
  </div>
</section>
</div>
<? include('./php/module/footer.php'); ?>
</body>
</html>

```

## nhCalculator.php

```

<?
  session_start();
  include("../config/conection.php");
?>

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet"
href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css" />

```

```

<script src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"></script>
<link rel="stylesheet" href="../../css/main.css">
<title>Document</title>
</head>

<body>
  <?php include('../../php/module/header.html'); ?>
  <div class="Site-content">
    <section>
      <div class="nhr-container container">
        <div id="map" class="pagePart"></div>
        <div class="form-container">
          <div id="msg"></div>

          <label for="fname"><b>Місце аварії</b></label>
          <select id="fname" name="fname">
            <option value="0" slected>Оберіть об'єкт на якому
сталася аварія</option>

            <?
            $factorys = mysqli_query($dbc, "SELECT f_id, fname
FROM factory");
            while ($factory = mysqli_fetch_assoc($factorys)) {
              <?>
              <option value="<? echo $factory['f_id']; ?>">
                <? echo $factory['fname']; ?>
              </option>

              <? } ?>
            </select>

            <label for="latitude"><b>Тип викинутої
речовини</b></label>
            <select id='substance' name="substance">
              <option value="0" slected>Оберіть тип викинутої
речовини</option>

              <?
              $enum_params = mysqli_fetch_assoc(mysqli_query($dbc,
"SHOW COLUMNS FROM tabledeptvalue WHERE Field = 'name_NHR'"));
              preg_match("/^enum\('(.*?)'\)$/",
$enum_params['Type'], $res);
              $enum = explode("'", "", $res['1']);

              for($i = 0; $i < count($enum); $i++){
                <?>
                <option value="<?echo $enum[$i]?>">
                  <?echo $enum[$i] ?>
                </option>
                <?
                }
              <?>
            </select>

            <label for="countSub"><b>Кількість викинутої речовини,
т</b></label>
            <input type="number" placeholder="Введіть кількість
викинутої речовини" name="countSub" required>

            <label for="wcount"><b>Відсоток працівників забезпечених
протигазами</b></label>

```

```

        <input type="number" placeholder="Введіть відсоток
працівників" name="wcount" required>

        <button id="calculateButton">Розрахувати</button>
    </div>
</div>
</section>
<section id="resultSection">
    <div class="nhr-container container">
        <div id="mapImg" class="pagePart"></div>
        <div id="resultTable" class="vertical"></div>
    </div>
    <div class="nhr-container container">
        <div id="damageTable" class="vertical"></div>
    </div>
</section>
</div>

<? include ('../php/module/footer.php'); ?>

<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.0/jquery.min.js'></scr
ipt>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/skycons/1396634940/skycons.min.js
'></script>
<script src="/js/helpers/readJSON.js"></script>
<script src="/js/calc/getWether.js"></script>
<script src="/js/getters/localGetter.js"></script>
<script src="/js/getters/serverGetter.js"></script>
<script src="/js/helpers/drawOnCanvas.js"></script>
<script src="/js/helpers/drawTable.js"></script>
<script src="/js/calc/workWithMap.js"></script>
<script src="/js/calc/parceCanvas.js"></script>
<script src="/js/lib/html2canvas.min.js"></script>
<script src="/js/calc/nhr.js"></script>
<script src="/js/module/nhr.js"></script>
</body>

</html>

```

## fireCalculatr.php

```

<?
    session_start();
    include("../config/conection.php");
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css" />
    <script src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"></script>
    <link rel="stylesheet" href="../../css/main.css">
    <title>Document</title>
</head>

```

```

<body>
  <?php include('../php/module/header.html'); ?>
  <div class="Site-content">
    <section>
      <div class="nhr-container container">
        <div id="map" class="pagePart"></div>
        <div class="form-container" style="height: 512px; overflow-y:
auto;">
          <div id="msg"></div>

          <label for="fname"><b>Варіант вводу координат
землетрусу</b></label>
          <select name="inputType" id="inputType">
            <option value="" disabled selected>--Оберіть варіант
вводу--</option>
            <option value="marker">Маркер на мапі</option>
            <option value="coordinates">Координати місця
винекнення</option>
          </select>

          <label for="latitude"><b>Широта</b></label>
          <input type="number" placeholder="Введіть широту"
name="latitude" required disabled>

          <label for="longitude"><b>Довгота</b></label>
          <input type="number" placeholder="Введіть довготу"
name="longitude" required disabled>

          <label for="fireType"><b>Вид пожежі</b></label>
          <select id="fireType" name="fireType">
            <option value="" disabled selected>--Оберіть вид
пожежі--</option>
            <option value="1">Верховий стійкий</option>
            <option value="2">Верховий побіжний</option>
            <option value="3">Низовий</option>
          </select>

          <label for="burnabilityClass"><b>Клас горимості лісових
насаджень</b></label>
          <select id="burnabilityClass" name="burnabilityClass">
            <option value="" disabled selected>--Оберіть клас
горимості--</option>
            <option value="1">Чисті і з домішкою листяних порід
хвойні насадження (крім модринових)</option>
            <option value="2">Чисті з домішкою хвойних порід
листяні насадження, а також модринові насадження</option>
          </select>

          <label for="avgHeghtCarbon"><b>Середня висота
нагару</b></label>
          <select id='avgHeghtCarbon' name="avgHeghtCarbon">
            <option value="" disabled selected>Оберіть середню
висоту нагару</option>

          <?
            $enum_params = mysqli_fetch_assoc(mysqli_query($dbc,
"SHOW COLUMNS FROM wood_damage WHERE Field = 'height'"));
            preg_match("/^enum\ (\'(.*)\'\)\$/",
$enum_params['Type'], $res);
            $enum = explode("'",',', $res['1']);

            for($i = 0; $i < count($enum); $i++){
?>

```

```

        <option value="<?echo $enum[$i]?>">
            <?echo $enum[$i] ?>
        </option>
    <?
    }
    ?>
</select>

    <label for="avgTreeDiameter"><b>Середній діаметр
деревостю</b></label>
    <select id='avgTreeDiameter' name="avgTreeDiameter">
        <option value="" disabled selected>Оберіть середній
діаметр деревостю</option>

        <?
        $enum_params = mysqli_fetch_assoc(mysqli_query($dbc,
"SHOW COLUMNS FROM wood_damage WHERE Field = 'diameter'"));
        preg_match("/^enum\('(.*?)'\)$/",
$enum_params['Type'], $res);
        $enum = explode("'", "", $res['1']);

        for($i = 0; $i < count($enum); $i++){
    ?>
        <option value="<?echo $enum[$i]?>">
            <?echo $enum[$i] ?>
        </option>
        <?
        }
    ?>
</select>

    <label for="startPerimeter"><b>Початковий периметр зони
загорання, м</b></label>
    <input type="number" placeholder="Введіть початковий
периметр загорання" name="startPerimeter" required>

        <button id="calculateButton">Позрахувати</button>
    </div>
</div>
</section>
<section id="resultSection">
    <div class="nhr-container container">
        <div id="mapImg" class="pagePart"></div>
        <div id="resultTable" class="vertical"></div>
    </div>
    <div class="nhr-container container">
        <div id="damageTable" class="vertical"></div>
    </div>
</section>
</div>

<? include('../php/module/footer.php'); ?>

<script
src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.0/jquery.min.js'></scr
ipt>
<script
src='https://cdnjs.cloudflare.com/ajax/libs/skycons/1396634940/skycons.min.js
'></script>
<script src="/js/helpers/readJSON.js"></script>
<script src="/js/calc/getWether.js"></script>
<script src="/js/helpers/readJSON.js"></script>

```

```

    <script src="/js/getters/serverGetter.js"></script>
    <script src="/js/helpers/drawOnCanvas.js"></script>
    <script src="/js/helpers/drawTable.js"></script>
    <script src="/js/lib/html2canvas.min.js"></script>
    <script src="/js/calc/workWithMap.js"></script>
    <script src="/js/module/forestFire.js"></script>
    <script src="/js/calc/forestfire.js"></script>
</body>

</html>

```

## earthquakeCalculator.php

```

<?
    session_start();
    include("../config/conection.php");
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css" />
    <script src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"></script>
    <link rel="stylesheet" href="../../css/main.css">
    <title>Document</title>
</head>

<body>
    <?php include('../php/module/header.html'); ?>
    <div class="Site-content">
        <section>
            <div class="nhr-container container">
                <div id="map" class="pagePart"></div>
                <div class="form-container">
                    <div id="msg"></div>

                    <label for="fname"><b>Варіант вводу координат
землетрусу</b></label>
                    <select name="inputType" id="inputType" required>
                        <option value="" disabled selected>--Оберіть варіант
вводу--</option>
                        <option value="marker">Маркер на мапі</option>
                        <option value="coordinates">Координати місця
визначення</option>
                    </select>

                    <label for="latitude"><b>Широта</b></label>
                    <input type="number" placeholder="Введіть широту"
name="latitude" required disabled>

                    <label for="longitude"><b>Довгота</b></label>
                    <input type="number" placeholder="Введіть довготу"
name="longitude" required disabled>

                    <label for="magnitude"><b>Магнітуда</b></label>
                    <input type="number" placeholder="Введіть магнітуду"
name="magnitude" required>

```

```

        <label for="depth"><b>Глибина гіпоцентру</b></label>
        <input type="number" placeholder="Введіть глибину
гіпоцентру" name="depth" required>

        <button id="calculateButton">Розрахувати</button>
    </div>
</div>
</section>
<section id="resultSection" style="display: none">
    <div class="nhr-container container">
        <div id="mapImg" class="pagePart"></div>
        <div id="resultTable" class="vertical"></div>
    </div>
    <div class="nhr-container container">
        <div id="damageTable" class="vertical"></div>
        <div id="lifeSupportSystems" class="vertical"></div>
    </div>
</section>
</div>

<? include ('../php/module/footer.php'); ?>

<script src="/js/helpers/readJSON.js"></script>
<script src="/js/getters/serverGetter.js"></script>
<script src="/js/helpers/drawOnCanvas.js"></script>
<script src="/js/helpers/drawTable.js"></script>
<script src="/js/lib/html2canvas.min.js"></script>
<script src="/js/calc/workWithMap.js"></script>
<script src="/js/module/earthquake.js"></script>
<script src="/js/calc/earthquake.js"></script>
</body>

</html>

```

## workWithMap.js

```

async function workWithMap(coordinates, scale) {
    let latitude = coordinates.latitude;
    let longitude = coordinates.longitude;

    map.setView([latitude, longitude], scale);

    let promise = new Promise((resolve, reject) => {
        setTimeout(() => resolve("готово!"), 1000)
    });
    let result = await promise;

    goFunc();

    return new Promise((resolve, reject) => {
        setTimeout(() => resolve("готово!"), 1000)
    });
}

const map = L.map('map').setView([50.452554, 30.561522], 5);
L.tileLayer('https://{s}.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png', {
    attribution: '&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
contributors'
}).addTo(map);

```

```

L.control.scale().addTo(map);

function goFunc() {
  html2canvas(document.querySelector("#map"), {
    allowTaint: true,
    useCORS: true,
  }).then(function (canvas) {
    let mapImg = document.querySelector('#mapImg');
    let cnvs = document.querySelector("canvas");
    if (cnvs) cnvs.remove();
    canvas.id = 'canvas';
    mapImg.appendChild(canvas);
  })
};

function drawOnMap(drawFuncName, color, canvas, latitude, longitude, scale,
radius, angle) {
  let ctx = canvas.getContext('2d');

  if( drawFuncName == 'drawArc'){
    let rad = pixelValue(latitude, radius * 1000, scale)

    drawArc(color, ctx, [canvas.width / 2, canvas.height / 2], rad);
  } else if (drawFuncName == 'drawEllipse') {
    let sPixelA = pixelValue(latitude, radius[0], scale);
    let sPixelB = pixelValue(latitude, radius[1], scale);

    drawEllipse(color, ctx, [canvas.width / 2, canvas.height / 2],
[sPixelA, sPixelB], angle);

    return;
  } else if (drawFuncName == 'drawFireZone'){
    let sPixelA = pixelValue(latitude, radius[0], scale);
    let sPixelB = pixelValue(latitude, radius[1], scale);

    drawFireZone(color, ctx, [canvas.width / 2, canvas.height / 2],
[sPixelA, sPixelB], angle);

    return;
  }
}

function metersPerPixel(latitude, zoomLevel) {
  let earthCircumference = 40075016.6;
  let latitudeRadians = latitude * (Math.PI / 180);
  return earthCircumference * Math.cos(latitudeRadians) / Math.pow(2,
zoomLevel + 8);
};

function pixelValue(latitude, meters, zoomLevel) {
  return meters / metersPerPixel(latitude, zoomLevel);
};

```

## parceCanvas.js

```

function unitArrayToREguralArray(unitArray) {
  let regularArray = [];
  let rgbaArray = [];

  for(let i = 0; i < unitArray.data.length; i+=4){

```



```

        regularArray.push([unitArray.data[i],unitArray.data[i+1],
unitArray.data[i+2], unitArray.data[i+3]]);
    }

    let counter = 0;

    for (let i = 0; i < unitArray.height; i++){
        let tmpArray = [];
        for (let j = 0; j < unitArray.width; j++){
            tmpArray.push(regularArray[j+counter]);
        }
        rgbaArray.push(tmpArray);
        counter += unitArray.width;
    }

    return rgbaArray;
}

function getData() {
    let c = document.getElementById("canvas");
    let ctx = c.getContext("2d");
    let data = ctx.getImageData(0, 0, c.width, c.height)

    let arryaOfRGBAValue = unitArrayToREGuralArray(data);
    return arryaOfRGBAValue;
}

function findPixelsInZone(data, zoneColor, pixelColor){
    let totalCounter = 0;
    let positionOfLastZonePixel;

    for (let row of data){
        let tmpCounter = 0;
        let countPixel = false;
        let addToTotalCount = false;
        for (let i = 0; i < row.length - 1; i++){
            if(row[i][0] === zoneColor.r && row[i][1] === zoneColor.g &&
row[i][2] === zoneColor.b){
                positionOfLastZonePixel = i;
            } else if(!countPixel && i - positionOfLastZonePixel === 1){
                countPixel = !countPixel;
            }

            if(countPixel){
                if(row[i][0] === pixelColor.r && row[i][1] === pixelColor.g
&& row[i][2] === pixelColor.b){
                    tmpCounter++;
                }
                if(row[i + 1][0] === zoneColor.r && row[i + 1][1] ===
zoneColor.g && row[i + 1][2] === zoneColor.b){
                    totalCounter += tmpCounter;
                    break;
                }
            }
        }
    }

    return totalCounter;
}

```

```

let zoom = 12;
const routerName = "dbrouter",
      height = 1, //          ВЫСОТА ОБВАЛОВАНИЯ          с
формы
      today = new Date(),
      accidentTime = today.getHours() + ":" + today.getMinutes(), //
время аварии автоматом
      place = {
//          инфа о местности          с
формы
      type: "відкрита",
      place: "Ліс",
      leng: 1,
    },
    zoneBorderColorRGB = {
      r: 255,
      g: 0,
      b: 0,
    },
    forestColorRGB = {
      r: 178,
      g: 194,
      b: 157,
    },
    waterColorRGB = {
      r: 144,
      g: 204,
      b: 203,
    },
    fieldColorRGB = {
      r: 222,
      g: 221,
      b: 190,
    };

async function calculate(id, nhrType, nhrQuantity, provGasMasl) {
  const factoryStr = await workWithBD("factory", routerName, {
    id,
  });
  const factoryArr = factoryStr.split("|");

  const latitude = +factoryArr[0], //    широта
        longitude = +factoryArr[1], //    долгота
        peopleNum = +factoryArr[2], //    количество людей на
объекте
        city = factoryArr[3]; //    город в котором находится
объект

  const wetherObjJSON = await getWetherData(latitude, longitude);
  const wetherObj = JSON.parse(wetherObjJSON);

  const temperature = wetherObj.currently.temperature, //
температура воздуха
        windSpeed = wetherObj.currently.windSpeed, //
скорость ветра
        windAzimut = wetherObj.currently.windBearing; //
угол/азимут ветра

  let cloudiness = wetherObj.currently.summary; //
облачность

```

```

cloudiness =
    cloudiness == "Ясно" ?
    cloudiness :
    cloudiness == "Хмарно" ?
    cloudiness :
    "Напівясно";

const timeOfDay = getTimeOfDay(accidentTime); //Время суток

const degreeOfVerticalStability = await workWithBD("vertical",
routerName, {
    timeOfDay,
    cloudiness,
    windSpeed,
}); //ступінь вертикальної стійкості

const tablDeptZone = await workWithBD("tabledeptvalue", routerName,
{
    nhrQuantity,
    nhrType,
    degreeOfVerticalStability,
    temperature,
    windSpeed,
}); //Гт - табличне значення глибини зони

const ksx = await workWithBD("nhr_cloud", routerName, {
    nhrType,
    height,
}); //Ксх - коефіцієнт, що враховує тип сховища і характеризує
зменшення глибини розповсюдження хмари НХР при виливі "у піддон"

const kzm = await workWithBD("koef_kzm", routerName, {
    degreeOfVerticalStability,
    palce: place["place"],
}); //Кзм - коефіцієнт зменшення глибини розповсюдження хмари НХР
для кожного 1 км довжини закритої місцевост

const deptDecr = getDeptDecr(place.leng, kzm); //Гзм - величина, на
яку зменшується глибина розповсюдження хмари НХР

const deptCalculatedZone = getDeptCalculatedZone(tablDeptZone, ksx,
deptDecr); //розрахункової глибини отримане значення Гр

const transferSpeed = await workWithBD("v_transfer", routerName, {
    windSpeed,
    degreeOfVerticalStability,
}); //W - швидкість переносу повітряних мас

const deptTransfer = getDeptTransfer(transferSpeed); //глибини
переносу повітряних мас Гп

const forecastDeptZone = getForecastDeptZone(
    deptCalculatedZone,
    deptTransfer
); //за фактичну прогнозовану глибину зони забруднення, тобто Гпзхз

//zoom = forecastDeptZone > 10 ? 11 : forecastDeptZone < 2 ? 13 :
zoom;

if (forecastDeptZone > 5)
    zoom = 11;
else if (forecastDeptZone < 2)
    zoom = 13;

```

```

const widthForecastZone = getWidthForecastZone(
    degreeOfVerticalStability,
    forecastDeptZone
); //Ширина прогнозованої зони хімічного забруднення (Шпзхз)

const F = getF(windSpeed); //Коефіцієнт Ф, який залежить від
швидкості вітру

const squarePossibleZone = getSquarePossibleZone(forecastDeptZone,
F); //Площа зони можливого хімічного забруднення (ЗМХЗ).const

const K = getK(degreeOfVerticalStability); //де К - коефіцієнт

const N = 4; //N - час, на який розраховується глибина ПЗХЗ
const squarePredictedZone = getSquarePredictedZone(K,
forecastDeptZone, N); //Площа прогнозованої зони хімічного забруднення
(ПЗХЗ)

const pollutionDuration = await workWithBD("pollution_duration",
routerName, {
    nhrType,
    height,
    temperature,
    windSpeed,
});

const percentOfLoss = await workWithBD("population_loss",
routerName, {
    place: place.type,
    provGasMasl,
});
const peopleLoss = getPeopleLoss(peopleNum, percentOfLoss);

const meterperPixel = metersPerPixel(latitude, zoom);

await workWithMap({
    latitude,
    longitude,
},
zoom
);

const semiaxisA = (forecastDeptZone * 1000) / 2;
const semiaxisB = (widthForecastZone * 1000) / 2;

const canvas = document.querySelector('#canvas');
drawOnMap('drawEllipse', zoneBorderColorRGB, canvas, latitude,
longitude, zoom, [semiaxisA, semiaxisB], windAzimut);

createTable('resultTable', [
    [
        "Тип НХР, кількість, тон",
        "Глибина ПЗХЗ, км",
        "Ширина ПЗХЗ, км",
        "Площа ПЗХЗ, км.кв.",
        "Площа МЗХЗ, км.кв.",
        "Тривалість уражаючої дії, хв",
        "Втрати людей, чол",
    ],
    [
        nhrType + ", " + nhrQuantity,
        forecastDeptZone,
    ]
]

```

```

        widthForecastZone,
        squarePredictedZone,
        squarePossibleZone,
        pollutionDuration,
        `Всього: ${peopleLoss.totalLoss}, смерт.-
        ${peopleLoss.death}, серед.- ${peopleLoss.moderateSeverity}, легкі-
        ${peopleLoss.lightSeverity}`
    ],
    ], 'Таблиця 1 - Результати оцінки хімічної обстановки');

    const data = getData();

    const fieldInZone = findPixelsInZone(data, zoneBorderColorRGB,
    fieldColorRGB);
    const forestInZone = findPixelsInZone(data, zoneBorderColorRGB,
    forestColorRGB);
    const waterInZone = findPixelsInZone(data, zoneBorderColorRGB,
    waterColorRGB);

    const fieldSqrt = getSqrt(meterperPixel, fieldInZone, data.length);
    const forestSqrt = getSqrt(meterperPixel, forestInZone,
    data.length);
    const waterSqrt = getSqrt(meterperPixel, waterInZone, data.length);

    const peopleLossDamage = getPeopleLossDamage(peopleLoss);

    const cityInfoStr = await workWithBD("agriculture", routerName, {
        city,
    });
    const cityInfo = cityInfoStr.split("|");

    const agricultureDamage = getAgricultureDamage(
        fieldSqrt,
        cityInfo[0],
        cityInfo[1]
    );

    const foresDamage = getForestDamage(forestSqrt, cityInfo[2]);

    const waterDamage = 900 * waterSqrt;

    createTable('damageTable', [
        [
            'Витрати на відшкодування життя та здоров'я
населення',
            'Збитки від порушення сільськогосподарських угідь',
            'Збитки від втрати деревини та інших лісових
ресурсів',
            'Збитки рибного господарства'
        ],
        [
            peopleLossDamage.toFixed(2),
            agricultureDamage.toFixed(2),
            foresDamage.toFixed(2),
            waterDamage.toFixed(2)
        ]
    ], 'Таблиця 2 - Результати оцінки прогнозованого еколого-
    економічного збитку')
}

```

getWether.js

```

const goButton = document.querySelector("#go");
const KEY = "199cb0b508acd58820ca7ed8fea3afb3";

function getWetherData(latitude, longitude){
  const
url=`https://api.darksy.net/forecast/${KEY}/${latitude},${longitude}?lang=uk
&units=auto`;

  return getJSONWether(url);
}

```

## forestfire.js

```

//const for calc
const V0 = 0.4, // V0 - скорость распространения пожара на равнине в
безветренную погоду, м/мин
k = 0.16, // k - коэффициент, учитывающий раздувающее влияние пламени
C = 3,
routerName = "dbrouter"; // C - коэффициент, учитывающий удельную
теплоемкость горючих материалов

const zoom = 11,
zoneBorderColorRGB = {
r: 255,
g: 0,
b: 0,
},
zoneBorderColorRGB2 = {
r: 0,
g: 0,
b: 255,
},
zoneBorderColorRGB3 = {
r: 255,
g: 255,
b: 0,
};

//variable
const hourOne = 24,
hourTwo = 48;

async function calculate(latitude, longitude, fireType, burnabilityClass,
avgHeghtCarbon, avgTreeDiameter, startPerimeter) {
const wetherObjJSON = await getWetherData(latitude, longitude);
const wetherObj = JSON.parse(wetherObjJSON);

console.log(wetherObj)

const temperature = wetherObj.currently.temperature, //
температура воздуха
windSpeed = wetherObj.currently.windSpeed, // скорость ветра
windAzimut = wetherObj.currently.windBearing, // угол/азимут ветра
humidity = wetherObj.currently.humidity; // влажность

let dewPoint = getDewPoint(temperature, humidity),
indicator = getIndicator(temperature, dewPoint),
wetherClass = getWetherClass(indicator),
linearFireFrontSpeed = getLinearFireFrontSpeed(fireType, windSpeed),
linearFireWingSpeed = getLinearFireWingSpeed(windSpeed),
linearFireRearSpeed = getLinearFireRearSpeed(windSpeed),

```

```

perimeterGainOne = getPerimeterGain(linearFireFrontSpeed, hourOne),
newPerimeterOne = getPerimeterInHours(startPerimeter, perimeterGainOne),
newSquareOne = getSquareInHours(newPerimeterOne),
perimeterGainTwo = getPerimeterGain(linearFireFrontSpeed, hourTwo),
newPerimeterTwo = getPerimeterInHours(startPerimeter, perimeterGainTwo),
newSquareTwo = getSquareInHours(newPerimeterTwo),
unusableWood = getUnusableWood(fireType, burnabilityClass);

let woodDamage = await workWithBD("wood_damage", routerName, {
  avgTreeDiameter,
  avgHeghtCarbon,
  burnabilityClass
});
let woodDamageCharJSON = await workWithBD("wood_damage_char", routerName, {
  woodDamage,
});

let woodDamageChar = JSON.parse(woodDamageCharJSON);

await workWithMap({
  latitude,
  longitude,
},
zoom
);

const canvas = document.querySelector('#canvas');

let radiusCircle = getRadius(startPerimeter);
let sCircle = Math.ceil((Math.PI * Math.pow(radiusCircle, 2)) / 10000);
drawOnMap('drawArc', zoneBorderColorRGB, canvas, latitude, longitude, zoom,
radiusCircle / 1000);

let radiusNewCircleOne = getRadius(newPerimeterOne);
let axisOne = getAxis(radiusNewCircleOne, linearFireFrontSpeed,
linearFireRearSpeed, hourOne)
let semiaxisOne = getSemiaxis(axisOne);
let semiordinatOne = radiusNewCircleOne;
drawOnMap('drawFireZone', zoneBorderColorRGB2, canvas, latitude, longitude,
zoom, [semiaxisOne, semiordinatOne], windAzimut);

let radiusNewCircleTwo = getRadius(newPerimeterTwo);
let axisTwo = getAxis(radiusNewCircleTwo, linearFireFrontSpeed,
linearFireRearSpeed, hourTwo)
let semiaxisTwo = getSemiaxis(axisTwo);
let semiordinatTwo = radiusNewCircleTwo;
drawOnMap('drawFireZone', zoneBorderColorRGB3, canvas, latitude, longitude,
zoom, [semiaxisTwo, semiordinatTwo], windAzimut);

createTable('resultTable', [
  ['Час',
  'Площа лісової пожежі, Га ',
  'Периметр лісової пожежі, км',
  ],
  [
  'На момент виявлення(червоний) ',
  sCircle,
  startPerimeter,
  ],
  [
  'Через добу після виявлення(синій) ',
  newSquareOne / 1000,

```

```

newPerimeterOne,
],
[
'Через дві доби після виявлення(жовтий)',
newSquareTwo / 1000,
newPerimeterTwo,
],
], 'Таблиця 1 - Розповсюдження пожежі в часових проміжках');

let res;
if (fireType = 3) {
res = woodDamageChar[0][0] + ' Відсоток непригодної деревини = ' +
woodDamageChar[0][1];
} else {
res = unusableWood;
}

createTable('damageTable', [
['Клас пожежної небезпеки',
'Швидкість фронту пожежі, м/с',
'Швидкість флангу пожежі, м/с',
'Швидкість тилу пожежі, м/с',
'Відсоток деревини непригодної для реалізації'
],
[
wetherClass,
linearFireFrontSpeed,
linearFireWingSpeed,
linearFireRearSpeed,
res
]
], 'Таблиця 2 - Загальні відомості про пожежу');
};

function getS(square) {
return square * 10000;
}

function getSemiaxis(axis) {
return axis / 2;
}

function getAxis(radiusCircle, linearFireFrontSpeed, linearFireRearSpeed,
hours) {
return radiusCircle * 2 + linearFireFrontSpeed * hours + linearFireRearSpeed
* hours;
}

function getRadius(perimeter) {
return perimeter / (2 * Math.PI);
}

function getDewPoint(temperature, humidity) {
const F = getF(temperature, humidity);
return (237.7 * F) / (17.27 - F);
}

function getF(temperature, humidity) {
return (17.23 * temperature) / (237.7 + temperature) + Math.log(humidity /
100);
}

```



```

function getIndicator(noonTemperature, dewPoint) {
return ((noonTemperature - dewPoint) * noonTemperature) * 14;
}

function getWetherClass(indicator) {
if (indicator <= 300)
return 1; //отсутствие опасности
else if (indicator > 300 && indicator <= 1000)
return 2; //малая пожарная опасность
else if (indicator > 1000 && indicator <= 4000)
return 3; //средняя пожарная опасность
else if (indicator > 4000 && indicator <= 12000)
return 4; //высокая пожарная опасность
else if (indicator >= 12000)
return 5; //чрезвычайная опасность
}

function getLinearFireFrontSpeed(fireType, windSpeed) {
switch (+fireType) {
case 1:
return 120;
case 2:
return 250;
case 3:
return Math.ceil(((V0 + k * windSpeed) * Math.pow(1 + (windSpeed /
Math.sqrt(Math.pow(windSpeed, 2) + Math.pow(C, 2))), 2)) * 30);
default:
return;
}
}

function getLinearFireWingSpeed(windSpeed) {
return Math.ceil((V0 + k * windSpeed) * 30);
}

function getLinearFireRearSpeed(windSpeed) {
return Math.ceil(((V0 + k * windSpeed) * Math.pow(1 - (windSpeed /
Math.sqrt(Math.pow(windSpeed, 2) + Math.pow(C, 2))), 2)) * 30);
}

function getPerimeterGain(linearFireFrontSpeed, hours) {
return 3.3 * linearFireFrontSpeed * hours;
}

function getPerimeterInHours(startPerimeter, perimeterGain) {
return Math.ceil(+startPerimeter + perimeterGain);
}

function getSquareInHours(newPerimeter) {
return Math.ceil(4 * Math.pow(10, -6) * Math.pow(newPerimeter, 2));
}

function getUnusableWood(fireType, burnabilityClass) {
if (+fireType == 2 && +burnabilityClass == 1) {
return 50;
} else if (+fireType == 2 && +burnabilityClass == 2) {
return 70;
} else if (+fireType == 3 && +burnabilityClass == 1) {
return 30;
} else if (+fireType == 3 && +burnabilityClass == 2) {
return 60;
} else {
return;
}
}

```

```

}
}

```

## earthquake.js

```

const routerName = "dbrouter",
    zoom = 10,

    epicentrZoneBorderColorRGB = {
        r: 242,
        g: 12,
        b: 157,
    },

    dangerZoneBorderColorRGB = {
        r: 255,
        g: 0,
        b: 0,
    },
    midleDangerZoneBorderColorRGB = {
        r: 255,
        g: 111,
        b: 0,
    };

async function calculate(latitude, longitude, magnitude, depth) {
    await workWithMap({
        latitude,
        longitude,
    },
    zoom
    );

    const canvas = document.querySelector('#canvas');
    drawOnMap('drawArc', epicentrZoneBorderColorRGB, canvas, latitude,
longitude, zoom, 1);
    drawOnMap('drawArc', dangerZoneBorderColorRGB, canvas, latitude,
longitude, zoom, 10);
    drawOnMap('drawArc', midleDangerZoneBorderColorRGB, canvas, latitude,
longitude, zoom, 20);

    let epicentrIntensity = getIntensityEpicentr(magnitude, depth),
        firstZone = getIntensity(magnitude, depth, 10),
        secondZone = getIntensity(magnitude, depth, 20);

    let epicentrEffect, firstEffect, secondEffect;

    if (epicentrIntensity < 1) {
        epicentrEffect = await workWithBD('earthquakeeffects',
routerName, {
            intensity: 1
        }),
        firstEffect = await workWithBD('earthquakeeffects',
routerName, {
            intensity: 1
        }),
        secondEffect = await workWithBD('earthquakeeffects',
routerName, {
            intensity: 1
        });
    }
}

```

```

    });
  } else {
    epicentrEffect = await workWithBD('earthquakeeffects',
routerName, {
      intensity: epicentrIntensity
    }),
    firstEffect = await workWithBD('earthquakeeffects',
routerName, {
      intensity: firstZone
    }),
    secondEffect = await workWithBD('earthquakeeffects',
routerName, {
      intensity: secondZone
    });
  }

createTable('resultTable', [
  ['',
    'Інтенсивність землетрусу ',
    'Наслідки землетрусу',
  ],
  [
    'В епіцентрі виникнення',
    epicentrIntensity,
    epicentrEffect,
  ],
  [
    'В радіусі 10км від епіцентру',
    firstZone,
    firstEffect,
  ],
  [
    'В радіусі 20км від епіцентру',
    secondZone,
    secondEffect,
  ],
], 'Таблиця 1 - Наслідки землетрусу в кожній зоні, в залежності від
інтенсивності на поверхні');

let deathArr = [
  [
    'Тип будівлі',
    'Втрати населення в епіцентрі, %',
    'Втрати населення в радіусі 10км, %',
    'Втрати населення в радіусі 20км, %',
  ]
];

let epicentrDeath,
    firstZoneDeath,
    secondZoneDeath;

if (epicentrIntensity < 7) {
  let epicentrDeathJSON = await workWithBD('earthquakekdeath',
routerName, {
    intensity: 0
  }),
  firstZoneDeathJSON = await workWithBD('earthquakekdeath',
routerName, {
    intensity: 0
  }),
  secondZoneDeathJSON = await
workWithBD('earthquakekdeath', routerName, {

```

```

        intensity: 0
    });

    epicentrDeath = JSON.parse(epicentrDeathJSON);
    firstZoneDeath = JSON.parse(firstZoneDeathJSON);
    secondZoneDeath = JSON.parse(secondZoneDeathJSON);
} else {
    let epicentrDeathJSON = await workWithBD('earthquakekdeath',
routerName, {
        intensity: epicentrIntensity
    }),
    firstZoneDeathJSON = await workWithBD('earthquakekdeath',
routerName, {
        intensity: firstZone
    }),
    secondZoneDeathJSON = await
workWithBD('earthquakekdeath', routerName, {
        intensity: secondZone
    });

    epicentrDeath = JSON.parse(epicentrDeathJSON);
    firstZoneDeath = JSON.parse(firstZoneDeathJSON);
    secondZoneDeath = JSON.parse(secondZoneDeathJSON);
};

for (let i in epicentrDeath) {
    deathArr.push([
        epicentrDeath[i][0],
        epicentrDeath[i][1],
        firstZoneDeath[i][1],
        secondZoneDeath[i][1]
    ])
};

createTable('damageTable', deathArr, 'Таблиця 2 - Втрати населення в
кожній зоні, в залежності від інтенсивності на поверхні, %');

let systemArr = [
    [,
        'Система',
        'Стійкість систем в епіцентрі, %',
        'Стійкість систем в радіусі 10км, %',
        'Стійкість систем в радіусі 20км, %',
    ]
];

let epicentrSystem,
    firstZoneSystem,
    secondZoneSystem;

if (epicentrIntensity < 7) {
    let epicentrSystemJSON = await workWithBD('earthquakeakesystem',
routerName, {
        intensity: 0
    }),
    firstZoneSystemJSON = await
workWithBD('earthquakeakesystem', routerName, {
        intensity: 0
    }),
    secondZoneSystemJSON = await
workWithBD('earthquakeakesystem', routerName, {
        intensity: 0
    });
};

```

```

        epicentrSystem = JSON.parse(epicentrSystemJSON);
        firstZoneSystem = JSON.parse(firstZoneSystemJSON);
        secondZoneSystem = JSON.parse(secondZoneSystemJSON);
    } else {
        let epicentrSystemJSON = await workWithBD('earthquakeakesystem',
routerName, {
            intensity: epicentrIntensity
        }),
        firstZoneSystemJSON = await
workWithBD('earthquakeakesystem', routerName, {
            intensity: firstZone
        }),
        secondZoneSystemJSON = await
workWithBD('earthquakeakesystem', routerName, {
            intensity: secondZone
        });

        epicentrSystem = JSON.parse(epicentrSystemJSON);
        firstZoneSystem = JSON.parse(firstZoneSystemJSON);
        secondZoneSystem = JSON.parse(secondZoneSystemJSON);
    };

    for (let i in epicentrSystem) {
        systemArr.push([
            epicentrSystem[i][0],
            epicentrSystem[i][1],
            firstZoneSystem[i][1],
            secondZoneSystem[i][1]
        ])
    };

    createTable('lifeSupportSystems', systemArr, 'Таблиця 3 - Стійкість
систем життєзабезпечення, %', 'Примітка: У чисельнику -% систем
життєзабезпечення, здатних до функціонуванню негайно, а в знаменнику -
після відновлювальних робіт в Протягом доби.');
```

```

}
```

```

function getIntensityEpicentr(magnitude, depth) {
    return Math.round(1.5 * magnitude - 3.5 * Math.log10(depth) + 3);
}
```

```

function getIntensity(magnitude, depth, radius) {
    return Math.round(1.5 * magnitude - 3.5 *
Math.log10(Math.sqrt(Math.pow(radius, 2) + Math.pow(depth, 2))) + 3);
}
```

