

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Центр післядипломної освіти

(повна назва)

Кафедра _____ програмної інженерії

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Програма для розрахунку систем вентиляції підземних споруд подвійного
призначення та ПРУ

(тема)

Виконав:

студент 4 курсу, групи ПЗПП-22-1

Тельний В.М.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного
забезпечення

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Програмна інженерія

(повна назва освітньої програми)

Керівник доц. Колесников Д.О.

(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри

(підпис)

З.В.Дудар

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Центр _____ післядипломної освіти
 Кафедра _____ програмної інженерії
 Рівень вищої освіти _____ перший (бакалаврський)
 Спеціальність _____ 121 – Інженерія програмного забезпечення
 Тип програми _____ Освітньо-професійна
 Освітня програма _____ Програма Інженерія
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Тельному Віталію Миколайовичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи _____ Програма для розрахунку систем вентиляції підземних споруд подвійного призначення та ПРУ _____

Затверджена наказом по університету від 17 червня 2024 р. № 588Ст

2. Термін подання студентом роботи до екзаменаційної комісії 20.07.2024 _____

3. Вихідні дані до роботи Розробити програму для розрахунку систем вентиляції підземних споруд подвійного призначення та ПРУ. _____

4. Перелік питань, що потрібно опрацювати в роботі

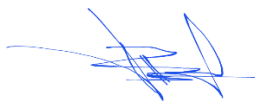
Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки. _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	20.05.2024	виконано
2	Створення специфікації ПЗ	22.05.2024	виконано
3	Проектування ПЗ	24.05.2024	виконано
4	Розробка ПЗ	28.05.2024	виконано
5	Тестування ПЗ	30.05.2024	виконано
6	Оформлення пояснювальної записки	05.06.2024	виконано
7	Підготовка презентації та доповіді	06.06.2024	виконано
8	Попередній захист	09.06.2024	виконано
9	Нормоконтроль, рецензування	06.06.2024	виконано
10	Здача роботи у електронний архів	10.06.2024	виконано
11	Допуск до захисту у зав. кафедри	11.06.2024	виконано

Дата видачі завдання 06 травня 2024р.

Студент (ка) _____
(підпис)



_____ Тельний В.М.

Керівник роботи _____
(підпис)

_____ доц. Колесников Д.О.
(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка: 79 с., 11 рисунків, 10 джерел.

БАЗА ДАНИХ SQLITE, МОВА ПРОГРАМУВАННЯ C#, СИСТЕМА ДЛЯ ПОБУДОВИ КЛІЄНТСЬКИХ ДОДАТКІВ, СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ, DB BROWSER, WPF

Метою кваліфікаційної роботи є розробка програми для розрахунку систем вентиляції підземних споруд подвійного призначення та ПРУ, з використанням реляційної бази даних та логіки об'єктно-орієнтованого програмування.

В ході створення додатку були використані середа розробки Visual Studio 2019, мова програмування C#, система побудови клієнтських додатків WPF, EntityFramework, база даних SQLite.

В результаті роботи був створений додаток, який дозволяє взаємодіяти та зберігати розрахунки для проектів будівництва. Також програма дозволяє шукати, фільтрувати інформацію, отримувати результати по конкретному приміщенні, а також генерує необхідні звіти в Word.

DATABASE MANAGEMENT SYSTEM, PROGRAMMING C#, DATABASE SQLITE, CLIENT APPENDIX SYSTEM, DB BROWSER, WPF

The purpose of the pre-certification practice is to develop a program for calculating the ventilation systems of dual purpose underground structures and PRU, using a relational database and the logic of object-oriented programming.

During the creation of the application, the development environment included Visual Studio 2019, C# language programming, WPF client application system, EntityFramework, SQLite database.

As a result, the work will create additions that allow interaction and save costs for everyday life projects. The program also allows you to search, filter information,

select results for a specific location, and also generate the necessary information in Word.

Я, Тельний Віталій Миколайович, студент гр. ПЗПП-22-1, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Програма для розрахунку систем вентиляції підземних споруд подвійного призначення та ПРУ», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Перелік скорочень.....	7
Вступ.....	8
1 Аналіз предметної галузі.....	9
1.1 Аналіз предметної галузі.....	9
1.2 Виявлення та вирішення проблем.....	17
1.3 Постановка задачі.....	19
2 Формування вимог до програмної системи.....	22
3 Архітектура та проектування програмного забезпечення.....	26
3.1 UML проектування ПЗ.....	26
3.2 Проектування архітектури ПЗ.....	30
3.3 Проектування структури зберігання даних.....	32
3.4 Приклади найцікавіших алгоритмів та методів.....	36
3.5 Створення UI/UX.....	42
4 Опис прийнятих програмних рішень.....	49
4.1 Вивантаження габаритів та переліку приміщень з Autodesk Revit ...	51
4.2 Виконання вентиляційних розрахунків.....	52
5 Тестування програмного забезпечення.....	63
Висновки.....	67
Перелік джерел посилання.....	68
Додаток А.....	69
Додаток Б.....	70
Додаток В.....	77

ПЕРЕЛІК СКОРОЧЕНЬ

- Entity – Framework
- SQL – Structured Query Language
- JSON – JavaScript Object Notation
- UML – Unified Modeling Language
- BLL – Business Logic Layer
- DAL – Data Access Layer
- SQLite – C-language library that implements a small, fast, self-contained, high-reliability
- UX – User Experience
- UI – User interface
- SDK – Serial Development Kit
- C# – Common Language Infrastructure
- WPF – Windows Presentation Foundation
- XML – Extensible Markup Language
- XAML – Extensible Application Markup Language
- PRU - Anti-radiation shelters

ВСТУП

У сучасному світі все прискорюється, а отже потреба у точних, кваліфікаційних, професійних рішеннях при будівництві набуває термін «на сьогодні, або навіть на вчора». Це ставить не аби який виклик перед сучасним інженером, адже якщо надати не зовсім вірну та занадто затягнуту за термінами інформацію, можна сказати з вірогідністю майже 100% що Замовник буде втрачений. Як інженер, який займається 19 років проектами, експертизами та впровадженням технічних рішень у будівництві, потребує у створенні спеціалізованої програми розрахунків.

Мета: Підвищення зручності обліку, ведення технічної документації та розрахунків для проектних закладів.

Завдання: Розробити інформаційну систему для ведення та оптимізації технічної документації та розрахунків при проектуванні будівель.

Дана система передбачає наступні етапи:

- провести аналіз та створити перелік необхідного функціоналу;
- визначити загальний архітектурний підхід та основні компоненти для побудови консольного застосунку;
- визначення технології запиту до баз даних.

Об'єкт дослідження: процес розрахунків та підключення баз даних у консольному додатку.

Предмет дослідження: застосування сучасних технологій створення консольного додатку для реалізації інженерних розрахунків.

Практична цінність: розроблено консольний додаток з використання сучасних технологічних рішень, що вирішує актуальні проблеми в сфері будівництва.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

Інженер є структурною частиною проектного інституту (організації), що забезпечує проектні види діяльності у будівельній галузі.

Проектний інститут об'єднує декілька відділів та підрозділів. Відділи поділяються на: будівельний (конструкції залізобетонні та металеві), інженерний (електропостачання, слаботочні системи, опалення, вентиляція і кондиціонування, водопостачання та каналізація), геодезичний, авторський надзор.

Відділами керує головний інженер підприємства та керівник, йому допомагають начальники відділів та підрозділів.

Мета роботи відділів – організація та підтримка проектного процесу роботи, до якого входять головні, ведучі та 1-3 категорії спеціалісти.

Робота відділів пов'язана:

- з організацією та підтримкою навчального процесу (для молодших спеціалістів);
- організація вивчення та застосування нормативної бази;
- організація виконання розрахунків;
- організація та виконання графічної частини проекту.

Далі будемо розглядати процес з організації та підтримки проектного процесу.

Головний спеціаліст виконує технічну роботу, а саме підтримує в актуальному стані нормативну інформацію, перелік проектів, склад робочих груп призначених до кожного об'єкту, перевірку та фіксацію випуску документації.

Процес проектування починається з формування інженерних та будівельних груп. Групи формуються на основі переліку штату та структури відділів. Перелік проектних об'єктів достатньо громіздка задача і може налічувати 10 або 20 будівель одночасно. Тому ведучі інженери потребують тут певної підтримки у розрахунках (майбутня задача автоматизації) від програмістів.

Під час формування розрахунків враховується інформація про місцезнаходження об'єкту, геодезичні та топографічні дані, кліматологію, типу будівництва (житлове, промислове, сільсько-господарське) та інше.

Інформаційні потреби інженерів:

- необхідна інформація про нові об'єкти, що потрапляє в проектний інститут за рахунок підписання договору на проектування, що надходить бухгалтерії у електронному вигляді (формат .txt);
- необхідні назви нормативних документів;
- необхідні вихідні дані нормативних документів;
- результати розрахунків.

Предметна галузь цього аналізу стосується систем вентиляції для підземних споруд подвійного призначення та протирадіаційних укриттів. Під підземними спорудами подвійного призначення маються на увазі будь-які підземні приміщення або структури, які можуть використовуватися як приміщення для людей та обладнання у звичайний час, а також як укриття в разі аварійних ситуацій або конфліктів.

Ця предметна галузь включає в себе розробку та проектування систем вентиляції, які забезпечують необхідну кількість свіжого повітря для людей, які перебувають в підземних умовах. Вона також охоплює аспекти вентиляції, що стосуються захисту від радіації та інших небезпечних речовин, які можуть бути присутніми в підземних спорудах під час аварійних ситуацій.

Основна мета цієї предметної галузі - забезпечити безпечні та комфортні умови для перебування людей у підземних спорудах у будь-яких обставинах, будь то нормальні умови експлуатації, екстрені ситуації або воєнний конфлікт.

Актуальність та значимість аналізу предметної галузі систем вентиляції для підземних споруд подвійного призначення та протирадіаційних укриттів полягають у кількох ключових аспектах:

- безпека населення: забезпечення ефективної системи вентиляції для підземних споруд подвійного призначення є критично важливим для

забезпечення безпеки населення у разі екстрених ситуацій, таких як природні катастрофи, терористичні атаки або воєнні дії;

- здоров'я та комфорт: відповідна вентиляція забезпечує достатню кількість свіжого повітря та відведення шкідливих викидів, що допомагає уникнути захворювань та забезпечує комфортні умови для перебування людей у підземних спорудах;
- національна безпека: розробка та вдосконалення систем вентиляції для протирадіаційних укриттів є ключовим аспектом національної безпеки в умовах можливої ядерної загрози або радіаційного зараження;
- військова стратегія: в армійському контексті, належна вентиляція є критично важливою для забезпечення ефективності та готовності підземних військових об'єктів, таких як командні пункти або військові бази;
- екологічна стійкість: ефективна система вентиляції може також допомогти у зменшенні впливу людської діяльності на довкілля, забезпечуючи ефективну рециркуляцію повітря та зменшення викидів шкідливих речовин у навколишнє середовище.

Отже, аналіз та подальший розвиток систем вентиляції для підземних споруд має велике значення для безпеки, здоров'я та комфорту населення, національної безпеки, військової стратегії та екологічної стійкості. Основні цілі аналізу включають такі завдання:

- виявлення проблемних місць: аналіз поточного стану систем вентиляції для підземних споруд та ідентифікація проблемних аспектів, які потребують уваги та вдосконалення;
- оцінка потреб користувачів: визначення потреб користувачів у відповідних і ефективних системах вентиляції, враховуючи їхні специфічні вимоги та умови експлуатації;
- аналіз сучасних технологій: огляд і оцінка сучасних технологій в області вентиляції для підземних споруд, щоб визначити можливості їхнього використання для вирішення проблем;

- визначення напрямків розвитку: встановлення ключових напрямків подальшого розвитку систем вентиляції, які мають на меті поліпшення ефективності, надійності та безпеки;
- формулювання вимог до програмного забезпечення: встановлення вимог до програмного забезпечення для розробки імплементації нових рішень у галузі вентиляції для підземних споруд;
- постановка завдань майбутньої розробки: визначення конкретних завдань та цілей, які повинні бути досягнуті у майбутніх розробках систем вентиляції для підземних споруд.

В сучасних системах вентиляції для підземних споруд застосовуються передові технології, такі як автоматизація, використання енергоефективних вентиляторів та розробка інтелектуальних систем керування.

Сучасний стан технологій вентиляції в підземних спорудах характеризується використанням передових розробок та інноваційних підходів для забезпечення належних умов життя, роботи та безпеки. Ось деякі з ключових аспектів сучасних технологій вентиляції в підземних спорудах:

- енергоефективність: сучасні системи вентиляції ставляться так, щоб бути енергоефективними, зменшуючи споживання електроенергії та покращуючи загальну ефективність;
- автоматизація: використання сучасних систем автоматизації дозволяє оптимізувати роботу вентиляційних систем у відповідь на змінні умови в підземних спорудах, що забезпечує ефективне та економне використання ресурсів;
- інтелектуальне керування: використання систем штучного інтелекту та алгоритмів машинного навчання дозволяє створити інтелектуальні системи керування, які можуть адаптуватися до змінних умов та забезпечувати оптимальну ефективність вентиляції;
- фільтрація та очищення повітря: сучасні системи вентиляції включають в себе високоефективні фільтри та системи очищення повітря, які

забезпечують видалення шкідливих частинок, бактерій та інших забруднювачів з повітря.

- моніторинг та контроль: використання сучасних систем моніторингу дозволяє постійно контролювати якість повітря, рівень вологості та інші параметри, що дозволяє оперативно реагувати на будь-які зміни та проблеми.
- інтеграція з іншими системами: сучасні системи вентиляції зазвичай інтегруються з іншими інженерними системами, такими як системи освітлення, системи безпеки та системи керування будівлею, для забезпечення синергії та максимальної ефективності.

Ці сучасні технології дозволяють створювати вискоелективні, надійні та безпечні системи вентиляції для підземних споруд, що відповідають найвищим стандартам якості та екологічної стійкості.

Попередні дослідження та розробки у сфері вентиляції протирадіаційних укриттів включають в себе різноманітні наукові та технічні ініціативи, спрямовані на забезпечення ефективного захисту людей у разі радіаційних аварій або конфліктів. Нижче відображені деякі з важливих аспектів цих досліджень та розробок:

- розробка фільтраційних систем: проведені дослідження включають в себе розробку та тестування ефективних систем фільтрації повітря, призначених для укриттів та бункерів, що можуть забезпечувати видалення радіоактивних частинок та забруднювачів з повітря;
- моделювання та аналіз вентиляційних систем: дослідницькі роботи у цій області включають в себе математичне моделювання та комп'ютерне симулювання вентиляційних систем у протирадіаційних укриттях для оцінки їх ефективності та оптимізації проектування;
- використання новітніх матеріалів: дослідження в цьому напрямку спрямовані на розробку та використання спеціальних матеріалів для вентиляційних систем, які можуть забезпечити високу стійкість до радіації та інших шкідливих факторів;

- аналіз ефективності заходів захисту: дослідження також включають аналіз ефективності різних заходів захисту, включаючи вентиляційні системи, з метою визначення їхнього впливу на рівень захисту людей від радіаційного випромінювання;
- стандартизація та регулювання: дослідження в області вентиляції протирадіаційних укриттів також сприяють розробці стандартів та нормативів, які регулюють проектування, будівництво та експлуатацію таких систем.

Ці попередні дослідження та розробки мають велике значення для подальшого розвитку та вдосконалення систем вентиляції в протирадіаційних укриттях, сприяючи підвищенню ефективності та безпеки захисту людей у разі радіаційних аварій.

Програма для розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів: Виявлення та вирішення проблем. Недоліки існуючих програмних рішень:

- відсутність адаптації до специфіки підземних споруд з подвійним призначенням та протирадіаційних укриттів;
- обмежений функціонал, який не враховує всі можливі сценарії експлуатації;
- неадекватна точність розрахунків в умовах підземного середовища.

Необхідність інтеграції нових технологій:

- використання штучного інтелекту для оптимізації розрахунків та прогнозування варіантів експлуатації вентиляційних систем;
- інтеграція з географічним інформаційними системами (GIS) для аналізу топографії місцевості та впливу на систему вентиляції.

Підвищення безпеки та надійності:

- розробка заходів забезпечення кібербезпеки для захисту від можливих кібератак;
- застосування резервування та реплікації даних для забезпечення надійності та уникнення втрати інформації.

Оптимізація інтерфейсу користувача:

- покращення інтерактивності та інтуїтивності інтерфейсу для забезпечення зручного та ефективного користування програмою;
- використання візуалізації даних для зручного аналізу результатів розрахунків.

Гнучкість та адаптабельність:

- розширення можливостей програми для врахування різних стандартів та нормативів у сфері вентиляції;
- забезпечення можливості індивідуальної настройки параметрів розрахунків залежно від потреб користувача.

Цей перелік проблем вказує на потенційні області для вдосконалення та розвитку програмного забезпечення для розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів. Розв'язання цих проблем може значно покращити якість та ефективність програми, що сприятиме підвищенню безпеки та комфорту у підземних умовах.

Існує кілька комерційних програм для розрахунку вентиляційних систем, які використовуються в інженерній практиці. Нижче наведено аналіз деяких з найпопулярніших програмних продуктів, їхні сильні та слабкі сторони.

Сильні сторони Autodesk Revit MEP [3]:

- інтеграція з BIM: Revit MEP (Mechanical, Electrical, and Plumbing) є частиною платформи Building Information Modeling (BIM), що дозволяє легко інтегрувати вентиляційні системи з іншими системами будівлі;
- візуалізація та документація: забезпечує високий рівень деталізації та візуалізації проектів, що спрощує процес створення документації та комунікації між командами;
- автоматизація процесів: програма автоматизує багато процесів проектування, включаючи розрахунок навантажень та вибір обладнання.

Слабкі сторони:

- висока вартість: Revit MEP є дорогою програмою, що може бути недоступною для малих компаній та індивідуальних інженерів;

- крива навчання вимагає значного часу для освоєння, особливо для користувачів, які не знайомі з BIM-технологіями.

Інший аналог - Carrier HAP (Hourly Analysis Program) [4]. Сильні сторони цього застосунку:

- точність розрахунків: Carrier HAP відомий своєю точністю у розрахунках теплових навантажень та вентиляційних потреб, що дозволяє створювати ефективні системи HVAC (Heating, Ventilation, and Air Conditioning);

- широкий спектр функцій: програма підтримує широкий спектр функцій для проектування та аналізу систем вентиляції та кондиціонування повітря.

Слабкі сторони:

- складність використання: інтерфейс може бути неінтуїтивним для нових користувачів, що ускладнює процес освоєння програми;

- обмежені можливості моделювання екстремальних умов: програма не завжди враховує специфічні вимоги для підземних споруд подвійного призначення та протирадіаційних укриттів.

Ще один аналог - ССК ТМ [5]. Його сильні сторони:

- модульна структура: ССК ТМ має модульну структуру, що дозволяє користувачам налаштовувати програму під специфічні потреби підбору обладнання;

- гнучкість: програма дуже гнучка та дозволяє підібрати різноманітне обладнання, включаючи системи вентиляції.

Слабкі сторони:

- високі вимоги до користувача: вимагає значних знань в галузі інженерії та програмування, що ускладнює її використання для новачків;

- обмежена підтримка стандартів: може мати обмежену підтримку специфічних національних та міжнародних стандартів, що ускладнює її застосування у деяких регіонах.

Розглянемо MS Excel [6]. Перевагами застосунку є:

- широкий набір функцій і формул: пропонує великий вибір вбудованих

функцій і формул для проведення різноманітних розрахунків, включаючи арифметичні операції, статистичні аналізи, фінансові розрахунки тощо;

- автоматизація розрахунків: можливість автоматично оновлювати результати розрахунків при зміні вихідних даних завдяки функціям і формулам, що значно зменшує ризик людської помилки;
- гнучкість у роботі з даними: дозволяє працювати з різними типами даних та легко їх перетворювати і формувати для подальших розрахунків;
- аналіз даних: інструменти для аналізу даних, такі як таблиці зведення, умовне форматування, функції пошуку і фільтрації, дозволяють легко аналізувати великі обсяги даних і знаходити необхідну інформацію;
- зручний інтерфейс: інтуїтивно зрозумілий інтерфейс користувача робить роботу з Excel простою і ефективною навіть для тих, хто не має глибоких знань у програмуванні або статистиці.

Недоліки MS Excel:

- обмеження у масштабованості: є призначений для роботи з дуже великими наборами даних. При роботі з мільйонами рядків і десятками колонок продуктивність може суттєво знижуватися;
- складність у відстеженні помилок: у формулах можуть бути важко помітні і діагностовані, особливо у великих та складних таблицях;
- обмеження у автоматизації: має обмеження у автоматизації процесів у порівнянні з більш спеціалізованими програмами для аналізу даних.

1.2 Виявлення та вирішення проблем

Недоліки існуючих програм для розрахунку вентиляції підземних споруд

Існуючі програми для розрахунку вентиляції підземних споруд мають ряд недоліків, які обмежують їх ефективність та зручність використання:

Обмежена функціональність:

- багато програм не враховують специфічні вимоги підземних споруд подвійного призначення та протирадіаційних укриттів;

- відсутність можливості моделювання різних аварійних сценаріїв та екстремальних умов.

Складність використання:

- деякі програми мають складний та неінтуїтивний інтерфейс, що ускладнює їх використання інженерами;
- високі вимоги до кваліфікації користувача для роботи з програмним забезпеченням.

Недостатня інтеграція з іншими системами:

- проблеми з інтеграцією програм для розрахунку вентиляції з іншими інженерними та архітектурними програмами;
- відсутність підтримки сучасних стандартів обміну даними.

Відсутність адаптивності:

- недостатня гнучкість у налаштуваннях та адаптації до специфічних вимог користувачів;
- відсутність можливості швидкого оновлення та розширення функціональності програм.

Низька надійність та продуктивність:

- проблеми з продуктивністю програм при обробці великих та складних проектів;
- недостатня надійність та стійкість до збоїв та помилок.

Академічні дослідження є важливим джерелом інновацій та нових ідей у галузі вентиляції підземних споруд. Вони дозволяють розробити та протестувати нові підходи, які можуть згодом бути впроваджені у комерційні продукти. У цьому розділі розглянемо останні наукові дослідження, перспективні підходи та інновації у сфері вентиляції підземних споруд.

Моделювання поведінки аерозолів та часток, що дозволяє передбачити поведінку аерозолів та інших шкідливих часток у підземних спорудах. Це включає моделювання розповсюдження диму під час пожежі, рух радіоактивних часток та інших небезпечних речовин. Такі моделі допомагають проектувати системи вентиляції, здатні ефективно видаляти ці забруднення.

Використання систем з рекуперацією енергії доказує, що системи вентиляції з рекуперацією енергії можуть значно знизити енергоспоживання, одночасно забезпечуючи належний рівень вентиляції. Це особливо важливо для підземних споруд, де ефективність енергоспоживання є критичним фактором.

Вплив кліматичних умов на ефективність вентиляції в різних кліматичних умовах дозволяє краще розуміти, як температура, вологість та інші параметри впливають на ефективність систем вентиляції. Це знання використовується для адаптації систем до різних географічних регіонів.

Дослідження акустичних властивостей вентиляційних систем також зосереджені на зниженні рівня шуму від вентиляційних систем, що є важливим для комфортного перебування людей у підземних спорудах. Використання нових матеріалів та конструктивних рішень допомагає знизити шумові характеристики систем.

1.3 Постановка задачі

Необхідно спроектувати та програмно реалізувати додаток розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів, а саме:

Розрахунок об'єму повітря:

- введення користувачем основних параметрів приміщення (об'єм, площа, висота стелі);
- введення користувачем параметрів зовнішнього середовища (температура, вологість);
- розрахунок необхідного об'єму повітря для вентиляції на основі введених даних.

Розрахунок витрат повітря:

- введення кількості людей, що перебувають у приміщенні;
- розрахунок необхідного об'єму повітря на людину відповідно до нормативних вимог;

- розрахунок загального об'єму повітря, необхідного для забезпечення комфортних умов.

Режими вентиляції:

- можливість вибору типу приміщення (підземне сховище, протирадіаційне укриття);
- розрахунок вентиляційної системи з урахуванням обраного режиму експлуатації.

Звітність:

- формування детального звіту з результатами розрахунків;
- можливість збереження звіту у форматах pdf або docx.

Інженер повинен мати можливість вказати тип розрахунку, кількість приміщень для обчислення, назву системи припливної вентиляції та назву витяжної системи вентиляції для кожного приміщення, також необхідність охолодження повітря у теплий період року.

Авторизований користувач додатку повинен мати доступ до баз даних, для внесення вихідних даних для розрахунків.

Програма повинна мати статистику виконання розрахунків певним інженером з можливістю фільтрації за датою та типом розрахунку.

Після виконання розрахунків Інженер отримує таблиці з зведеними розрахунками по кожному приміщенню та системі.

Для спрощення роботи інженерів під час роботи з інформацією вони повинні мати можливість:

- сортувати нормативні документ, за датою випуска, назвою, організацією та країною яка видає цей норматив;
- здійснювати пошук інформації про нормативні документи, вихідні дані, параметри приміщень;
- здійснювати фільтрацію інформації з результатів розрахунків, датою, типом розрахунку.

Інженери для спрощення аналізу інформації з проектного процесу повинні мати можливість отримання наступної статистики:

- про приміщення (номер, назва, площа, висота, рівень);
- про розрахунки (дата розрахунку, тип, id інженера, який виконує розрахунки);
- про нормативні документи (дата випуску, назва, зміни, актуальність).

Системи вентиляції підземних споруд подвійного призначення, таких як бомбосховища, ядерні сховища або підземні парковки, мають критичне значення для забезпечення безпеки, комфорту та здоров'я людей, що перебувають в таких умовах. Ефективна програма для розрахунку систем вентиляції в таких умовах дозволить інженерам та дизайнерам оптимізувати роботу вентиляційних систем та забезпечити високий рівень безпеки та комфорту.

Цільовою аудиторією програми для розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів є:

- інженери з вентиляції: фахівці, які відповідають за проектування, розробку та експлуатацію систем вентиляції підземних споруд.
- архітектори та дизайнери: спеціалісти, які включають вентиляційні системи в проекти підземних споруд та визначають їхні параметри.
- спеціалісти з безпеки: особи, які відповідають за забезпечення безпеки та відповідність нормам у сфері вентиляції підземних споруд.
- студенти та дослідники: особи, які вивчають та проводять дослідження в галузі вентиляції підземних споруд.
- керівництво підприємств та організацій: особи, які відповідають за прийняття рішень щодо встановлення та експлуатації вентиляційних систем у своїх підприємствах та організаціях.

Цей перелік представляє основні категорії користувачів, які будуть зацікавлені в використанні програми для розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів.

Програма призначена для інженерів та дизайнерів, які працюють у галузі будівництва та проектування підземних споруд. Вона має забезпечити ефективність, точність та безпеку в роботі з вентиляційними системами у підземних умовах.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Програмна система для розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів (ПРУ) має бути розроблена для забезпечення ефективного і безпечного функціонування даних об'єктів. Вимоги до програмної системи включають технічні, функціональні, не функціональні вимоги та вимоги до інтерфейсу користувача.

Наведемо функціональні вимоги. В першу чергу звернемо увагу на розрахунок параметрів вентиляції:

- розрахунок об'єму повітря, необхідного для вентиляції приміщення, з урахуванням кількості людей, обладнання та теплових навантажень;
- розрахунок кратності повітрообміну для різних типів приміщень;
- розрахунок температурного режиму, відносної вологості та інших параметрів мікроклімату.

Моделювання вентиляційних систем:

- моделювання різних типів вентиляційних систем (природна, механічна, комбінована);
- моделювання роботи вентиляційних систем у нормальному та аварійному режимах;
- врахування спеціальних вимог до вентиляційних систем ПРУ (фільтрація повітря, захист від радіоактивного та хімічного забруднення).

База даних обладнання:

- інтеграція з базою даних вентиляційного обладнання (вентилятори, фільтри, клапани тощо);
- можливість додавання, редагування та видалення записів у базі даних;
- автоматичний вибір оптимального обладнання на основі розрахованих параметрів.

Аналіз ефективності систем вентиляції:

- аналіз ефективності вентиляційних систем за різними критеріями (енергоспоживання, ефективність очищення повітря, надійність);
- генерація звітів та рекомендацій для оптимізації систем вентиляції.

Імпорт та експорт даних:

- імпорт даних з інших систем (Revit);
- експорт результатів розрахунків у різні формати (txt).

Вимоги до архітектури системи:

- модульна архітектура, що забезпечує легкість модифікації та розширення функціональності;
- використання сучасних мов програмування (C#) та фреймворків (Entity).

Вимоги до сумісності:

- сумісність з операційної системи версії Windows 7 і вище;
- інтеграція з популярними CAD-системами (Revit 2022 і вище).

Вимоги до сховища даних:

- використання надійної системи управління базами даних (SQLite);
- забезпечення резервного копіювання та відновлення даних.

Вимоги до продуктивності:

- забезпечення високої продуктивності при обробці великих об'ємів даних та складних розрахунків (одночасне обчислення будівлі з 300 приміщеннями);
- оптимізація алгоритмів для забезпечення мінімального часу розрахунків (не більше 5 сек).

Безпека має бути забезпечена з точки зору:

- захисту даних користувачів та конфіденційної інформації;
- використання сучасних методів шифрування та автентифікації.

Вимоги до інтерфейсу користувача - використовувати WPF. Застосунок має мати інтуїтивно зрозумілий інтерфейс - простий та зрозумілий інтерфейс, що дозволяє користувачам швидко освоїти систему та слід використовувати сучасні принципи дизайну UI/UX.

Користувач має мати:

- можливість налаштування інтерфейсу під індивідуальні потреби користувача;
- збереження налаштувань користувача для забезпечення зручності роботи.

Візуалізація даних:

- візуалізація результатів розрахунків у вигляді графіків, діаграм та схем;
- інтерактивні елементи для більш детального аналізу даних.

Мова інтерфейсу:

- підтримка багатомовності (українська, англійська);
- можливість швидкого перемикання між мовами.

Підтримка користувачів:

- інтеграція з системою допомоги та підтримки користувачів (онлайн-довідка, технічна підтримка);
- можливість отримання консультацій та допомоги безпосередньо через програму.

Виділимо нефункціональні вимоги.

Надійність:

- висока надійність та стабільність роботи системи;
- захист від збоїв та аварійних ситуацій.

Масштабованість:

- можливість масштабування системи для підтримки кількості користувачів (до 10) та обробки великих об'ємів даних (до 5 Мб);
- гнучкість у додаванні нових функцій та модулів.

Портативність:

- можливість роботи на різних платформах та пристроях (десктопні та мобільні версії);
- використання хмарних технологій для забезпечення доступу до даних з будь-якого місця.

Підтримка стандартів:

- відповідність міжнародним стандартам у галузі вентиляції та енергетичного аналізу будівель;
- використання стандартних форматів файлів для обміну даними.

Документування:

- наявність повної документації для користувачів та розробників;

- регулярне оновлення документації у разі змін у програмному забезпеченні.

Розробка програмної системи для розрахунку систем вентиляції підземних споруд подвійного призначення та ПРУ є складним і багатогранним процесом, що вимагає врахування багатьох аспектів. Визначення чітких та детальних вимог до системи є ключовим етапом, що забезпечує успішну реалізацію проекту. Запропонована система повинна бути надійною, гнучкою та зручною у використанні, забезпечуючи високий рівень точності та ефективності розрахунків.

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 UML проєктування ПЗ

«Unified Modeling Language (UML) - це стандарт для візуалізації, опису та документування архітектури програмних систем» [6]. UML містить діаграми, що допомагають розробникам та архітекторам програмного забезпечення проєктувати системи на різних рівнях абстракції. Для нашої системи ми будемо використовувати декілька основних діаграм UML:

3.1.1 Діаграма випадків використання (Use Case Diagram)

Діаграма випадків використання (Use Case Diagram) показує взаємодію користувачів із системою, зокрема які функції вони можуть виконувати. Для нашої програми, яка призначена для розрахунку систем вентиляції підземних споруд, діаграма випадків використання включатиме наступні елементи:

Користувачі:

- інженер - основний користувач системи, який виконує розрахунки вентиляції та зберігає дані;
- адміністратор - користувач, який управляє доступом до системи та її конфігурацією.

Випадки використання:

- розрахунок вентиляції - виконання необхідних розрахунків для забезпечення належної вентиляції підземних споруд;
- зберігання даних - збереження результатів розрахунків та інших важливих даних у базі даних;
- управління користувачами - адміністрування облікових записів користувачів, їх прав доступу та інших параметрів.

Після детального аналізу була створена Use-case діаграма (див. рис. 3.1).

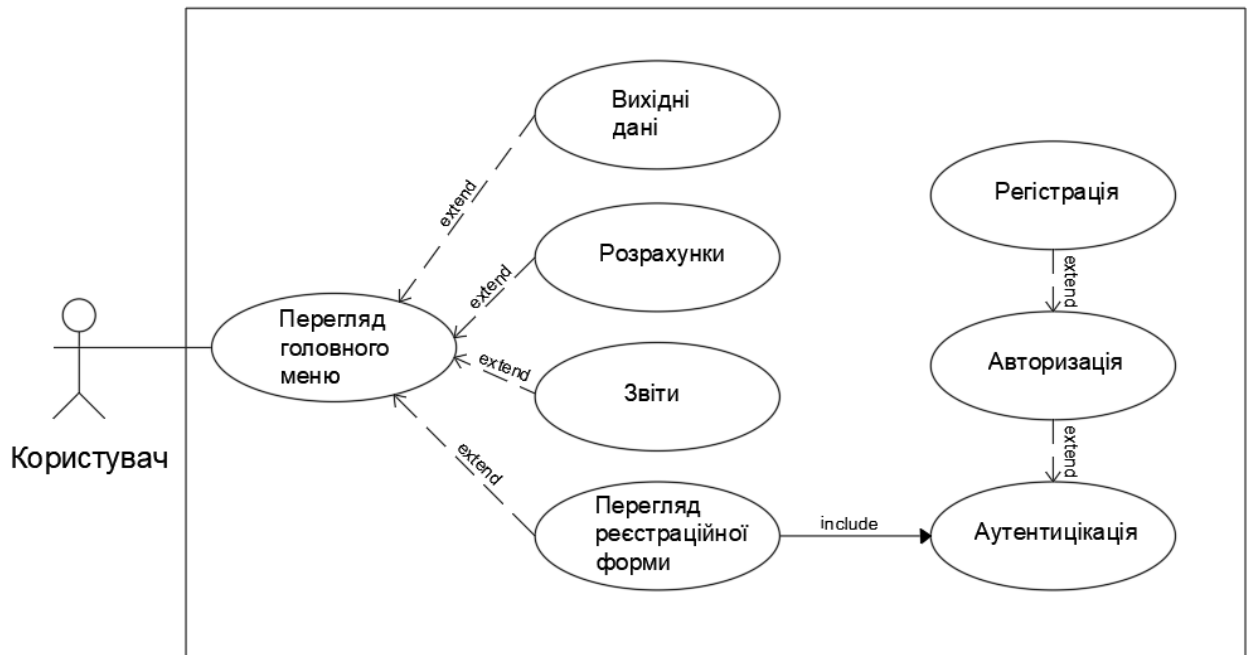


Рисунок 3.1 – Use-case діаграма

Діаграма випадків використання надає загальне уявлення про те, як різні користувачі взаємодіють із системою. Вона допомагає визначити основні функції системи та взаємодії між ними, що є важливим кроком у процесі проєктування програмного забезпечення.

3.1.2 Діаграма класів (Class Diagram)

Діаграма класів показує структуру системи на рівні класів та їх взаємодії. Вона містить основні класи, їх атрибути, методи та зв'язки між ними. Для нашої системи основні класи включають:

- клас Вентиляція (Ventilation);
- клас Система зберігання (StorageSystem);
- клас Користувач (User);
- клас Розрахунки (Calculations).

Діаграма класів відображена на рис. 3.2

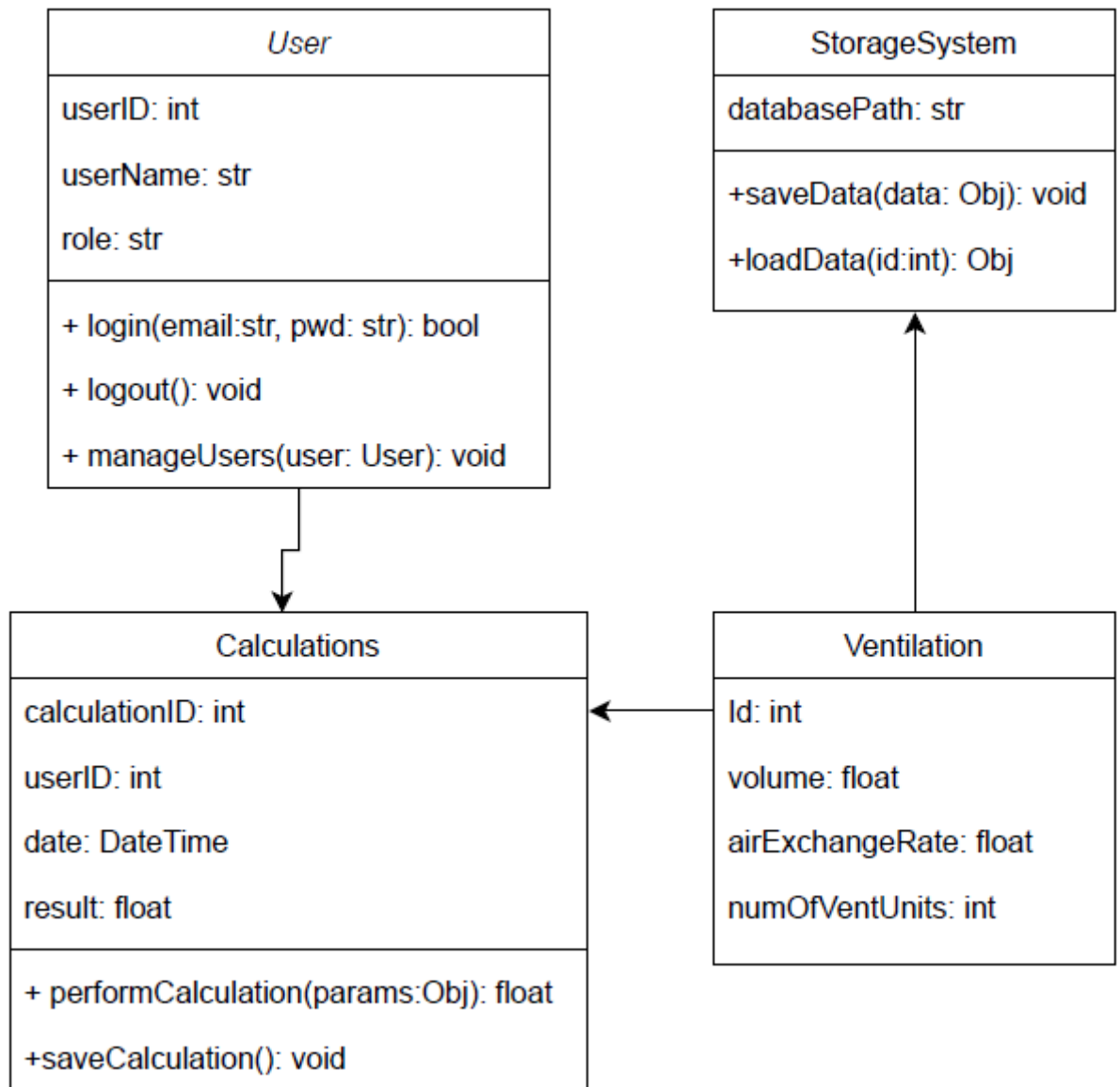


Рисунок 3.2 – Діаграма класів

Опис зв'язків між класами

- **User** та **Calculations**: один користувач може мати багато розрахунків, тому між ними зв'язок один до багатьох;
- **Calculations** та **StorageSystem**: розрахунки зберігаються в системі зберігання даних. Розрахунок використовує методи системи зберігання для збереження та завантаження даних;
- **Ventilation** та **Calculations**: клас вентиляції використовує методи для розрахунків повітрообміну та визначення кількості вентиляційних установок. Результати зберігаються в класі розрахунків.

Діаграма класів показує структуру системи на рівні класів та їх взаємодії. Це допомагає зрозуміти, як різні компоненти системи пов'язані між собою та яким чином вони обмінюються даними. Це важливий крок у проєктуванні програмного забезпечення, який забезпечує чітке уявлення про структуру та функціональність системи.

3.1.3 Діаграма послідовностей (Sequence Diagram)

Діаграма послідовностей показує взаємодію об'єктів у хронологічному порядку для виконання певного випадку використання. У нашому прикладі розглянемо послідовність розрахунку вентиляції.

Учасники:

- користувач;
- інтерфейс користувача (UI);
- контролер;
- система розрахунків (Calculations);
- система зберігання (StorageSystem);
- база даних (Database).

Послідовність розрахунку вентиляції:

- користувач вводить дані;
- система перевіряє дані;
- система виконує розрахунки;
- система зберігає результати;
- система повертає результати користувачу.

Діаграма послідовностей відображена на рис.

Опишемо докладно етапи:

- користувач вводить необхідні параметри для розрахунку вентиляції в інтерфейсі користувача (UI);
- інтерфейс користувача передає введені дані до розрахунків;
- обчислювальний модуль перевіряє коректність введених даних;

- якщо дані некоректні, модуль повертає повідомлення про помилку в інтерфейс користувача;
- якщо дані коректні, модуль передає їх до системи розрахунків;
- система розрахунків (Calculations) приймає дані від контролера;
- виконує необхідні розрахунки для вентиляції;
- повертає результати обчислювальному модулю;
- система розрахунків передає результати до системи зберігання (Database);
- система зберігання зберігає результати у базу даних;
- база даних підтверджує успішне збереження даних системі зберігання;
- система зберігання повертає підтвердження обчислювальному модулю.
- обчислювальний модуль передає результати розрахунків та підтвердження збереження інтерфейсу користувача.
- інтерфейс користувача відображає результати користувачу.

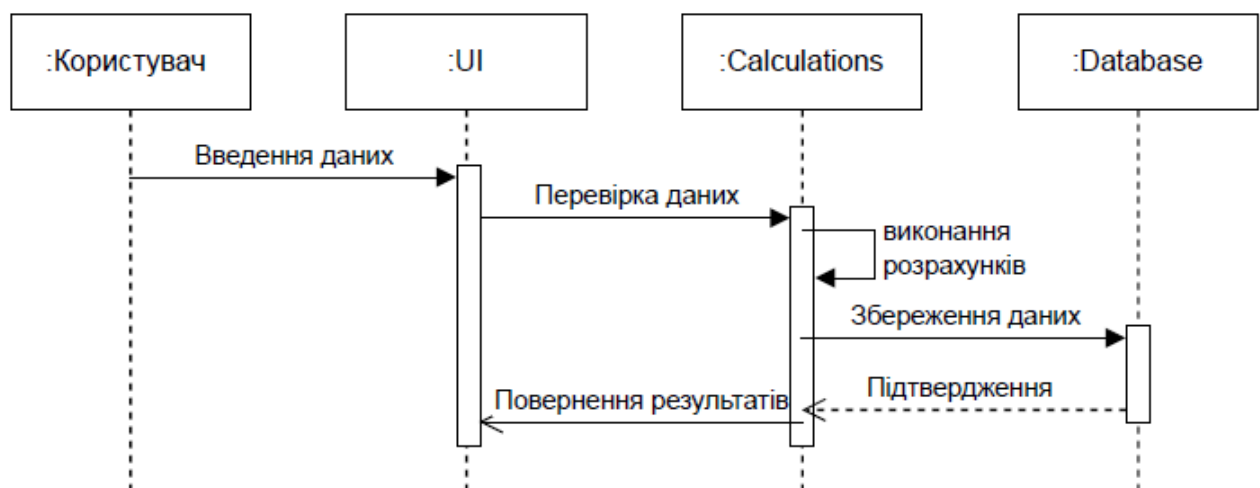


Рисунок 3.3 – Діаграма послідовностей

3.2 Проектування архітектури ПЗ

Програма побудована на базі архітектури додатку і включає в себе 2 головних компонента:

- WPF головний інтерфейс взаємодії адміністратора з інформаційною системою;
- SQLite – база даних, в якій зберігається необхідна інформація [7].

Архітектурну діаграму проекту можна побачити на рисунку 3.4

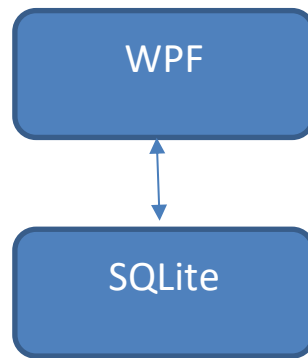


Рисунок 3.4 – Архітектурна діаграма інформаційної системи

Під час проектування програмного забезпечення для розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів, використовується модульний підхід до архітектури системи.

Модульний підхід передбачає розбиття програмного забезпечення на невеликі самостійні модулі, кожен з яких виконує певну функцію або набір функцій. Це дозволяє забезпечити високу ступінь розширюваності, підтримки та обслуговування системи.

Взаємодія між модулями здійснюється через визначений інтерфейс, який визначає, як модулі можуть взаємодіяти один з одним. Це може бути реалізовано за допомогою API (Application Programming Interface), який визначає доступні функції та їхні параметри для взаємодії з іншими модулями.

Переваги модульного підходу відображені далі:

- модульність: програмне забезпечення легко розширювати та модифікувати, оскільки кожен модуль є окремим та незалежним;
- підтримка: легка підтримка та обслуговування, оскільки зміни в одному модулі не впливають на інші частини системи;
- тестування: можливість проводити тестування окремих модулів незалежно від інших, що полегшує виявлення та виправлення помилок;
- перевикористання коду: можливість використовувати модулі в інших проектах або системах, що підвищує ефективність розробки.

Модульний підхід до архітектури системи програмного забезпечення дозволяє забезпечити високу розширюваність, підтримку та обслуговування системи. Взаємодія між модулями здійснюється через визначений інтерфейс, що дозволяє їм співпрацювати та виконувати потрібні завдання.

Архітектура програмного забезпечення визначає структуру системи на високому рівні та описує основні компоненти та їх взаємодію. Для нашої програми оберемо архітектуру з трьома основними компонентами: інтерфейс користувача (UI), бізнес-логіка (BLL).

Інтерфейс користувача буде реалізований за допомогою WPF, що дозволяє створювати багатофункціональні графічні інтерфейси.

Бізнес-логіка міститиме основні алгоритми та методи для розрахунку вентиляції, обробки даних та інших функцій.

Шар доступу до даних буде відповідати за взаємодію з базою даних SQLite, зберігання та витягування даних.

Бізнес-логіка (Business Logic Layer, BLL) є центральною частиною архітектури програмного забезпечення, яка містить основні алгоритми та методи для розрахунку вентиляції, обробки даних та інших функцій. Вона визначає, як дані трансформуються та передаються між інтерфейсом користувача (UI).

Основні компоненти бізнес-логіки:

- клас VentilationCalculator;
- клас UserManagement;
- клас Validation.

3.3 Проектування структури зберігання даних

Реляційна база даних – це база даних, яка використовується для зберігання та організації доступу до взаємопов'язаних елементів інформації. Реляційні бази даних ґрунтуються на реляційній моделі – інтуїтивно зрозумілому, наочному поданні інформації у вигляді таблиць. Кожен рядок у таблиці такої бази даних являє собою запис унікальним ідентифікатором – ключем. Стовпці таблиць мають атрибути даних, що дає змогу встановлювати зв'язки між елементами даних. Саме

цьому, реляційна модель бази даних є найбільш ефективною для вирішення задачі даного курсового проекту (дивись рисунок 3.5).

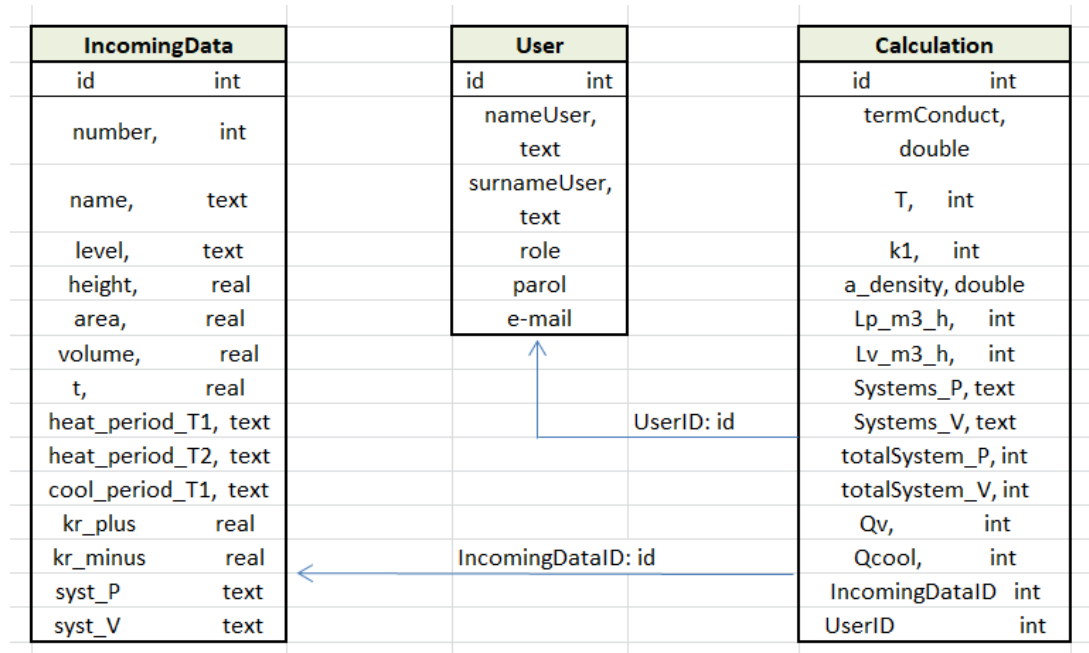


Рисунок 3.5 – Схема реляційної бази даних

Нормальна форма – це вимога, що висувається до структури таблиць в теорії реляційних баз даних для усунення з бази зайвих функціональних залежностей між розрахунками. В нашому проекті при нормалізації виконується приведення таблиць до третьої нормальної форми. Відношення знаходиться у третій нормальній формі, коли воно знаходиться у першій та другій нормальній формі, тобто всі атрибути атомарні, кожний не ключовий атрибут повністю функціонально залежить від первинного ключа відношення та між неключовими атрибутами немає транзитивних залежностей.

В усіх відношеннях всі атрибути є атомарними, тому вони знаходяться в першій нормальній формі (НФ).

При зведенні до другої нормальної форми було доведено, що всі відношення не містять неповних функціональних залежностей, тобто в складі потенційного ключа відсутня менша підмножина атрибутів від якої можна також вивести дану функціональну залежність.

При зведенні до третьої нормальної форми була проведена перевірка на відсутність транзитивних функціональних залежностей. Отже, отримана схема БД знаходиться в 3НФ.

Для зберігання даних у нашій системі використовується SQLite, яка є легкою і не вимагає серверного оточення. Проектування структури зберігання даних включає визначення таблиць, зв'язків між ними та атрибутів.

Основні таблиці в базі даних:

- користувачі (Users);
- розрахунки (Calculations);
- дані для розрахунків (InputData).

Таблиця Users містить інформацію про користувачів системи, включаючи їх ідентифікатор, ім'я, email та роль. Атрибути:

- id (INTEGER, PRIMARY KEY, AUTOINCREMENT): унікальний ідентифікатор користувача;
- name (TEXT, NOT NULL): ім'я користувача;
- email (TEXT, NOT NULL): email користувача;
- role (TEXT, NOT NULL): роль користувача (наприклад, інженер, адміністратор).

Скрипт на створення таблиці:

```
CREATE TABLE Users (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name TEXT NOT NULL,  
    email TEXT NOT NULL,  
    role TEXT NOT NULL  
);
```

Таблиця Розрахунки (Calculations) містить результати розрахунків, виконаних користувачами, та інформацію про те, хто і коли виконав розрахунок. Атрибути:

- id (INTEGER, PRIMARY KEY, AUTOINCREMENT): унікальний ідентифікатор розрахунку;

- user_id (INTEGER, NOT NULL, FOREIGN KEY): ідентифікатор користувача, який виконав розрахунок;
- date (TEXT, NOT NULL): дата виконання розрахунку;
- result (REAL, NOT NULL): результат розрахунку.

Скрипт на створення таблиці:

```
CREATE TABLE Calculations (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL,
    date TEXT NOT NULL,
    result REAL NOT NULL,
    FOREIGN KEY (user_id) REFERENCES Users(id)
);
```

Таблиця Дані для розрахунків (InputData) містить вхідні параметри для кожного розрахунку, що дозволяє відстежувати, які дані були використані для отримання конкретного результату. Атрибути:

- id (INTEGER, PRIMARY KEY, AUTOINCREMENT): унікальний ідентифікатор вхідних даних;
- calculation_id (INTEGER, NOT NULL, FOREIGN KEY): ідентифікатор розрахунку;
- parameter1 (REAL, NOT NULL): перший параметр розрахунку;
- parameter2 (REAL, NOT NULL): другий параметр розрахунку;
- parameterN... (REAL, NOT NULL): інші параметри розрахунку.

Скрипт на створення таблиці:

```
CREATE TABLE InputData (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    calculation_id INTEGER NOT NULL,
    parameter1 REAL NOT NULL,
    parameter2 REAL NOT NULL,
    -- Додайте інші параметри за потреби
    FOREIGN KEY (calculation_id) REFERENCES Calculations(id)
);
```

Опишемо зв'язки між таблицями:

- таблиця Users має зв'язок "один-до-багатьох" з таблицею Calculations, оскільки один користувач може виконувати багато розрахунків;

- таблиця Calculations має зв'язок "один-до-багатьох" з таблицею InputData, оскільки один розрахунок може мати багато вхідних параметрів.

Проектування структури зберігання даних є критично важливим етапом при розробці програмного забезпечення. Використання таблиць Users, Calculations та InputData забезпечує ефективне зберігання даних, необхідних для роботи системи, дозволяючи легко отримувати та аналізувати інформацію про користувачів, розрахунки та вхідні параметри. Правильне визначення зв'язків між таблицями забезпечує цілісність даних та спрощує доступ до них.

3.4 Приклади найцікавіших алгоритмів та методів

Під час розробки програмного забезпечення для розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів важливим є розробка алгоритмів та моделей, що визначають ефективність та параметри цих систем.

Розробка алгоритмів розрахунку повітрообміну та вибору обладнання:

- розрахунок потреб у повітрі: розробка алгоритмів для визначення кількості повітря, необхідного для задоволення потреб у вентиляції підземних споруд в різних умовах експлуатації.
- вибір обладнання: розробка алгоритмів для вибору оптимального обладнання (вентиляторів, фільтрів, воздуховодів тощо) для забезпечення ефективної роботи систем вентиляції.
- моделювання гідродинаміки: розробка алгоритмів для моделювання гідродинамічних процесів, що відбуваються в системі вентиляції, з урахуванням різноманітних факторів, таких як турбулентність потоку, опори воздуховодів тощо.

Моделі для оцінки ефективності роботи системи в різних режимах

- моделі теплообміну: розробка моделей для оцінки теплообміну між повітрям та оточуючим середовищем у підземних спорудах з урахуванням теплових навантажень та характеристик системи вентиляції.

- моделі шуму: розробка моделей для оцінки рівня шуму, що генерується системою вентиляції, з урахуванням параметрів обладнання та геометрії підземних споруд.
- моделі енергоефективності: розробка моделей для оцінки енергоефективності роботи систем вентиляції в різних режимах експлуатації та для ідентифікації можливостей зниження споживання енергії.

Один з ключових алгоритмів системи - це розрахунок вентиляції для підземних споруд:

- визначення обсягу приміщення;
- обрахунок необхідного повітрообміну;
- визначення кількості вентиляційних установок.

Алгоритм оптимізації розміщення вентиляційних установок використовує методи оптимізації для визначення оптимального розташування вентиляційних установок з метою забезпечення рівномірного повітрообміну.

Алгоритм розрахунку вентиляції для підземних споруд. Цей алгоритм визначає необхідний обсяг повітря для забезпечення адекватного повітрообміну, а також кількість вентиляційних установок, необхідних для цього. Основні кроки алгоритму:

- визначення обсягу приміщення
- обрахунок необхідного повітрообміну
- визначення кількості вентиляційних установок

Для розрахунку вентиляції спочатку необхідно визначити обсяг приміщення, яке потребує вентиляції. Це може бути обчислено за формулою:

$$\text{Обсяг приміщення} = \text{Довжина} \times \text{Ширина} \times \text{Висота}$$

Фрагмент коду для розрахунку об'єму приміщення:

```
public double CalculateVolume(double length, double width, double height)
```

```

{
    return length * width * height;
}

```

Наступний крок - обрахунок необхідного повітрообміну. Необхідний обсяг повітря для обміну розраховується за допомогою обсягу приміщення та коефіцієнта обміну повітря (кількість разів, коли повітря має бути замінено за певний час, наприклад, за годину):

Необхідний повітрообмін = Обсяг приміщення × Коефіцієнт обміну повітря

Фрагмент коду для розрахунку повітрообміну:

```

public double CalculateAirExchange(double volume, double
airExchangeRate)
{
    return volume * airExchangeRate;
}

```

Останній крок - визначення кількості вентиляційних установок, необхідних для забезпечення необхідного повітрообміну. Це можна зробити, розділивши необхідний повітрообмін на потужність однієї вентиляційної установки і округливши результат у більшу сторону для врахування неповних установок:

*Кількість вентиляційних установок =
= Необхідний повітрообмін / Потужність установки*

Фрагмент коду для розрахунку кількості вентиляційних установок:

```

public int DetermineVentUnits(double volume, double airExchangeRate,
double unitCapacity)
{
    double totalAirExchange = CalculateAirExchange(volume,
airExchangeRate);
    return (int) Math.Ceiling(totalAirExchange / unitCapacity);
}

```

Розглянемо приклад, як можна використати ці методи разом для виконання розрахунку вентиляції підземного приміщення:

```
public void PerformVentilationCalculation(double length, double width,
double height, double airExchangeRate, double unitCapacity)
{
    VentilationCalculator calculator = new VentilationCalculator();

    // Крок 1: Визначення обсягу приміщення
    double volume = calculator.CalculateVolume(length, width, height);

    // Крок 2: Обрахунок необхідного повітрообміну
    double requiredAirExchange =
calculator.CalculateAirExchange(volume, airExchangeRate);

    // Крок 3: Визначення кількості вентиляційних установок
    int ventUnits = calculator.DetermineVentUnits(volume,
airExchangeRate, unitCapacity);

    Console.WriteLine($"Обсяг приміщення: {volume} м³");
    Console.WriteLine($"Необхідний повітрообмін: {requiredAirExchange}
м³/год");
    Console.WriteLine($"Кількість вентиляційних установок:
{ventUnits}");
}
```

Алгоритм розрахунку вентиляції для підземних споруд є критично важливим компонентом нашої системи. Він включає три основні кроки: визначення обсягу приміщення, обрахунок необхідного повітрообміну та визначення кількості вентиляційних установок. Правильне впровадження цього алгоритму забезпечує надійну та ефективну вентиляцію, що є важливим для безпеки та комфорту користувачів.

Алгоритм оптимізації розміщення вентиляційних установок використовує методи оптимізації для визначення оптимального розташування вентиляційних установок з метою забезпечення рівномірного повітрообміну. Цей підхід враховує геометрію приміщення, об'єми повітря та інші параметри, що впливають на ефективність вентиляції. Основні етапи алгоритму:

- збір вихідних даних: збираються дані про розміри приміщення, кількість вентиляційних установок, розташування перешкод тощо;
- моделювання приміщення: створюється математична модель приміщення, яка враховує всі особливості просторової геометрії;

- визначення цільової функції: визначається цільова функція, яка оцінює ефективність вентиляції в залежності від розміщення установок;
- застосування методу оптимізації: використовується один з методів оптимізації (наприклад, генетичний алгоритм, метод градієнтного спуску або симуляційного відпалу) для пошуку оптимального розташування вентиляційних установок;
- оцінка та корекція результатів: отримані результати оцінюються, і, за необхідності, алгоритм корегує розміщення установок для досягнення оптимального повітрообміну.

Для демонстрації розглянемо реалізацію спрощеного алгоритму оптимізації розміщення вентиляційних установок за допомогою методу симуляційного відпалу.

```
using System;

public class VentilationOptimizer
{
    private double[,] room; // Матриця, що представляє приміщення
    private int numInstallations; // Кількість вентиляційних установок
    private Random random;

    public VentilationOptimizer(double[,] room, int numInstallations)
    {
        this.room = room;
        this.numInstallations = numInstallations;
        this.random = new Random();
    }
}
```

Цільова функція, що оцінює рівномірність повітрообміну:

```
private double ObjectiveFunction(int[] positions)
{
    double score = 0.0;
    // Простий критерій оцінки (наприклад, сума відстаней від
    центрів приміщень)
    foreach (int pos in positions)
    {
        int x = pos / room.GetLength(1);
        int y = pos % room.GetLength(1);
        score += CalculateAirflow(x, y);
    }
    return score;
}
```

Функція, що моделює потік повітря (простий приклад):

```
private double CalculateAirflow(int x, int y)
{
    // Для простоти, враховуємо відстань до центра приміщення
    int centerX = room.GetLength(0) / 2;
    int centerY = room.GetLength(1) / 2;
    return Math.Sqrt((x - centerX) * (x - centerX) + (y - centerY)
* (y - centerY));
}
```

Основний алгоритм симуляційного відпалу:

```
public int[] SimulatedAnnealing(double initialTemperature, double
coolingRate)
{
    // Випадкове початкове розташування установок
    int[] currentPositions = new int[numInstallations];
    for (int i = 0; i < numInstallations; i++)
    {
        currentPositions[i] = random.Next(room.Length);
    }

    int[] bestPositions = (int[])currentPositions.Clone();
    double bestScore = ObjectiveFunction(currentPositions);

    double temperature = initialTemperature;

    while (temperature > 1)
    {
        // Створення нового сусіднього стану
        int[] newPositions = (int[])currentPositions.Clone();
        int pos = random.Next(numInstallations);
        newPositions[pos] = random.Next(room.Length);

        double currentScore = ObjectiveFunction(currentPositions);
        double newScore = ObjectiveFunction(newPositions);

        // Приймаємо новий стан з певною ймовірністю
        if (AcceptanceProbability(currentScore, newScore,
temperature) > random.NextDouble())
        {
            currentPositions = newPositions;
        }

        // Оновлюємо найкраще знайдене рішення
        if (newScore < bestScore)
        {
            bestPositions = (int[])newPositions.Clone();
            bestScore = newScore;
        }
    }
}
```

Зниження температури:

```

    temperature *= coolingRate;
    }

    return bestPositions;
}

```

Функція для визначення ймовірності прийняття нового стану:

```

private double AcceptanceProbability(double currentScore, double
newScore, double temperature)
{
    if (newScore < currentScore)
    {
        return 1.0;
    }
    return Math.Exp((currentScore - newScore) / temperature);
}
}

```

Алгоритм оптимізації розміщення вентиляційних установок допомагає визначити найкраще розташування для забезпечення рівномірного повітрообміну в приміщенні. Використання методів оптимізації, таких як симуляційний відпал, дозволяє ефективно знаходити оптимальні рішення для складних завдань вентиляції.

3.5 Створення UI / UX або іншого дизайну системи;

Створення інтуїтивного та ефективного інтерфейсу користувача є ключовим для успіху програмного забезпечення. Дотримання основних принципів UI/UX дизайну, таких як простота, мінімізація кроків та використання зрозумілих символів, забезпечує легкість та зручність використання системи. Інтерфейси, розроблені з урахуванням цих принципів, дозволять користувачам швидко та ефективно виконувати необхідні завдання, забезпечуючи при цьому високий рівень задоволеності від роботи з системою.

Головне вікно програми буде розроблено таким чином, щоб забезпечити зручний доступ до всіх основних функцій системи. Вікно міститиме меню навігації, поля для введення даних, кнопки для запуску розрахунків та поля для відображення результатів. Елементи головного вікна:

- меню навігації: дозволяє користувачам швидко переміщуватися між основними розділами програми (Файл, Довідка тощо);
- поля для введення даних: знаходяться в центральній частині вікна і дозволяють вводити необхідні параметри для розрахунків;
- кнопки для запуску розрахунків: дозволяють користувачам запускати процес розрахунку вентиляційних параметрів;
- поля для відображення результатів: відображають результати розрахунків у зрозумілому та структурованому вигляді.

Після запуску BuildingSystems.exe можна потрапити у наступне вікно відображене на рисунки 3.6.

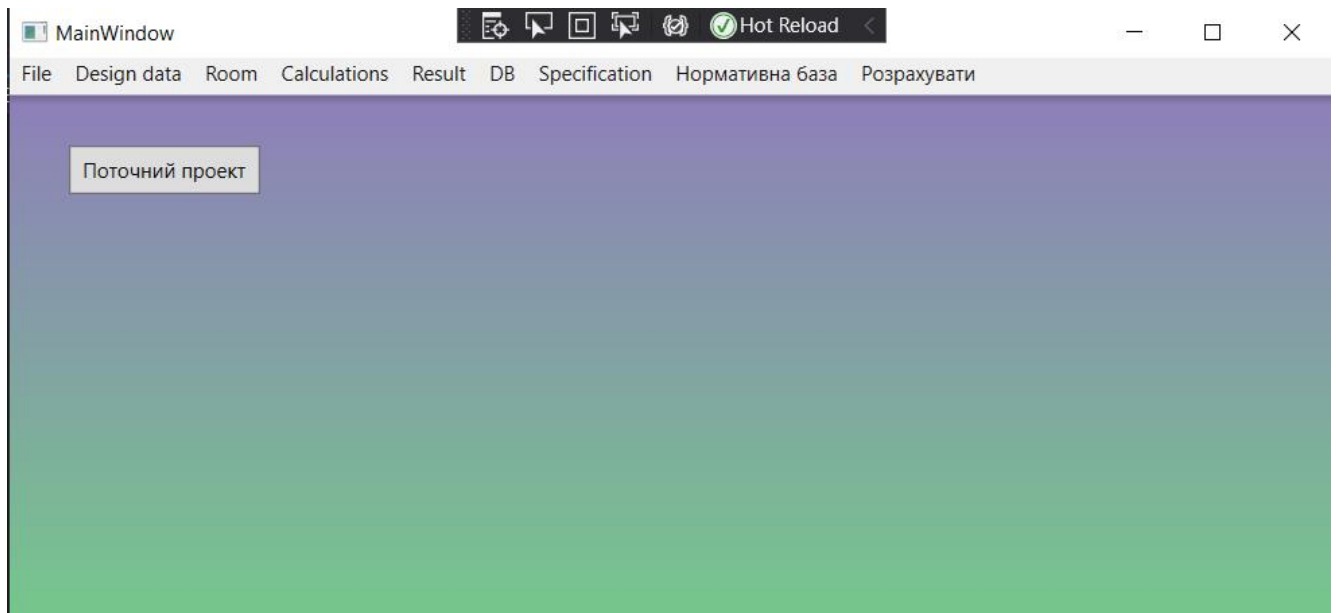


Рисунок 3.6 – Загальний вид програми

Далі необхідно перевірити, який поточний проект зараз завантажений у додаток. Для цього необхідно натиснути на кнопку «Поточний проект».

У вікні додатку відображеному на рисунку 3.7 можна отримати необхідну інформацію.

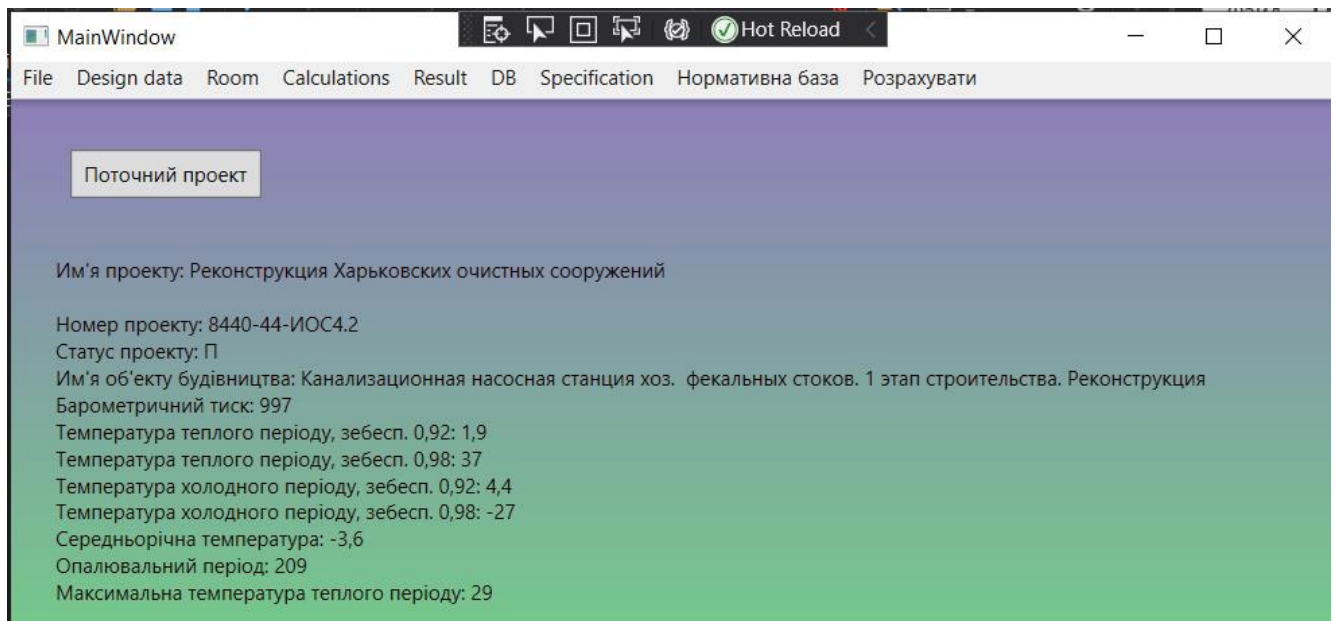


Рисунок 3.7 – Загальний вид програми з вихідними даними

Наступним кроком необхідно вибрати тип розрахунку. На даному пункту меню відображено розрахунок всіх приміщень одразу або по одинці. Поточна інформація відображена на рисунку 3.8.

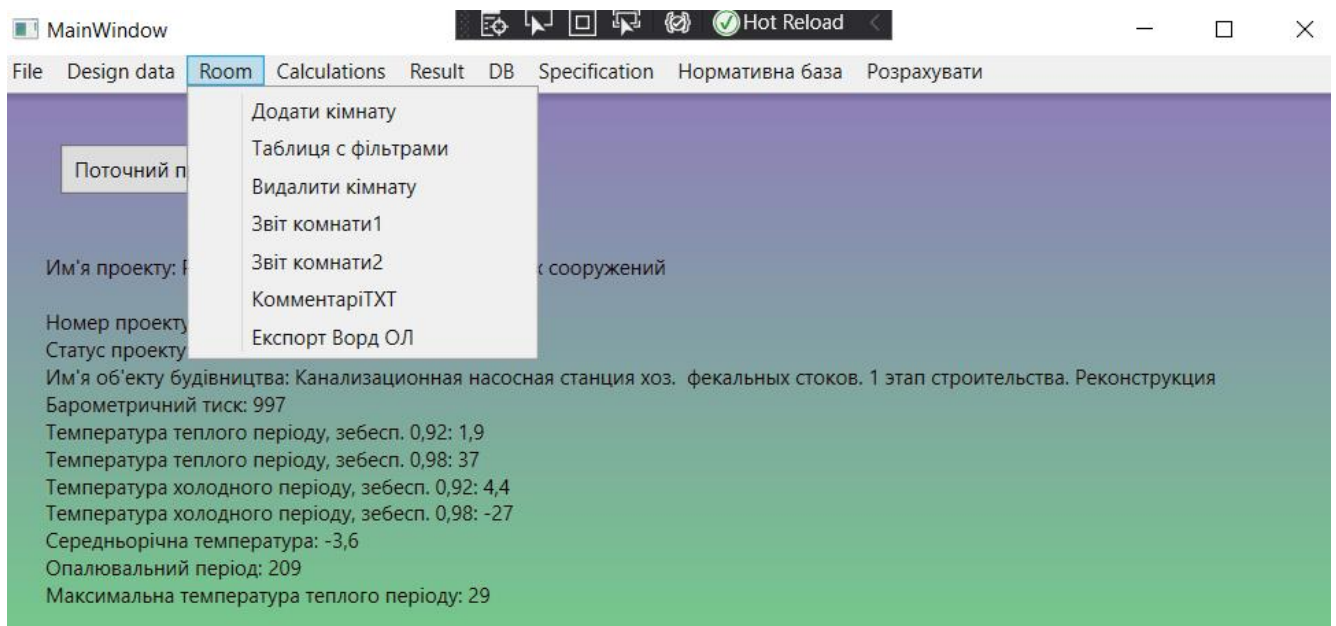


Рисунок 3.8 – Загальний вид програми. Вибір типу розрахунків

Надалі можна отримати відображення результатів розрахунків які розміщені у базах даних. Наприклад запит №1 (кількість запитів не обмежена). Відображено на рисунку 3.9.

Після перегляду, аналізу та перевірки можна закрити додаток. Інформація зберігається автоматично під час виконання розрахунків

Рисунок 3.9 – Загальний вид програми. Відображення запиту №1

Приклад XAML розмітки для головного вікна:

```
<Window x:Class="VentilationApp.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Ventilation System" Height="600" Width="800">
  <Grid>
    <!-- Меню навігації -->
    <Menu VerticalAlignment="Top">
      <MenuItem Header="Файл">
        <MenuItem Header="Відкрити"/>
        <MenuItem Header="Зберегти"/>
      </MenuItem>
      <MenuItem Header="Довідка"/>
    </Menu>

    <!-- Основний вміст -->
    <StackPanel VerticalAlignment="Top" Margin="20,50,20,20">
      <!-- Поля для введення даних -->
      <GroupBox Header="Введення даних" Margin="0,10,0,10">
        <StackPanel Margin="10">
          <TextBox x:Name="RoomDimensions"
PlaceholderText="Розміри приміщення (Д x Ш x В)" Margin="0, 5"/>
          <TextBox x:Name="VentilationParameters"
PlaceholderText="Параметри вентиляції" Margin="0, 5"/>
        </StackPanel>
      </GroupBox>
    </StackPanel>
  </Grid>
</Window>
```

```

        <!-- Кнопки для запуску розрахунків -->
        <StackPanel Orientation="Horizontal"
HorizontalAlignment="Center" Margin="0,10,0,10">
            <Button Content="Запуск розрахунків" Width="200"
Height="40" Margin="5" Click="Розрахувати_Click"/>
            <Button Content="Очистити" Width="200" Height="40"
Margin="5" Click="Очистити_Click"/>
        </StackPanel>

        <!-- Поля для відображення результатів -->
        <GroupBox Header="Результати розрахунків"
Margin="0,10,0,10">
            <StackPanel Margin="10">
                <TextBlock x:Name="ResultsText" Text="Результати
будуть відображені тут" Margin="0, 5"/>
            </StackPanel>
        </GroupBox>
    </StackPanel>
</Grid>
</Window>

```

Пояснення компонентів головного вікна

Меню навігації

– розташоване у верхній частині вікна, містить елементи меню "Файл" та "Довідка";

– підпункти меню "Файл" включають опції "Відкрити" та "Зберегти".

Поля для введення даних

– розміщені в групі "Введення даних";

– містять два текстові поля: для введення розмірів приміщення та параметрів вентиляції.

Кнопки для запуску розрахунків

– розташовані горизонтально під полями введення даних;

– кнопка "Запуск розрахунків" запускає процес розрахунку;

– кнопка "Очистити" очищує введені дані.

Поля для відображення результатів

– розміщені в групі "Результати розрахунків";

– включають текстовий блок, який відображає результати розрахунків.

Головне вікно програми для розрахунку систем вентиляції підземних споруд розроблено з урахуванням основних принципів UI/UX дизайну, забезпечуючи простоту та інтуїтивність використання. Всі основні елементи, такі як меню

навігації, поля для введення даних, кнопки для запуску розрахунків та поля для відображення результатів, розташовані логічно та зручно для користувача.

Форми для введення даних є критично важливим елементом будь-якого програмного забезпечення. Вони забезпечують зручний і точний введення параметрів для розрахунків. Для мінімізації помилок користувачів і покращення зручності використання, важливо використовувати різні контрольні елементи, такі як випадаючі списки, чекбокси та радіокнопки.

Основні принципи дизайну форм для введення даних:

- зрозумілість і простота (користувачі повинні легко розуміти, яку інформацію потрібно ввести, всі поля мають бути чітко підписані);
- мінімізація помилок (використання контрольних елементів (випадаючі списки, чекбокси, радіокнопки) для обмеження можливих варіантів вибору, валідація введених даних в режимі реального часу);
- ергономіка (поля для введення мають бути розташовані в логічній послідовності, важливі параметри повинні бути розташовані у верхній частині форми).

Поле введення розмірів приміщення:

- три текстові поля для введення довжини, ширини та висоти приміщення;
- поля розташовані горизонтально для зручності введення.

Поле введення кількості вентиляційних установок:

- випадаючий список (ComboBox) для вибору кількості вентиляційних установок;
- обмеження вибору допомагає уникнути введення невірних даних.

Поле вибору типу вентиляційної установки:

- випадаючий список для вибору типу вентиляційної установки (Приточна, Витяжна, Комбінована);
- використання зрозумілих термінів допомагає користувачам швидко зробити вибір.

Поле вибору режиму роботи вентиляції:

- два радіокнопки для вибору режиму роботи вентиляції (Автоматичний, Ручний);
- обмеження вибору до одного з двох варіантів допомагає уникнути помилок.

Чекбокс для вибору додаткових опцій:

- чекбокс для вибору режиму аварійної вентиляції;
- додаткові опції надають користувачам більше можливостей для налаштування системи.

Кнопка для запуску розрахунків:

- кнопка для запуску процесу розрахунків вентиляційних параметрів;
- розташована внизу форми для логічного завершення процесу введення даних.

Дизайн форм для введення даних забезпечує зручний та ефективний процес введення параметрів для розрахунків. Використання контрольних елементів, таких як випадаючі списки, чекбокси та радіокнопки, допомагає мінімізувати помилки користувачів та підвищує зручність використання системи. Форматування та розташування елементів форми відповідає основним принципам UI/UX дизайну, забезпечуючи інтуїтивний та логічний процес введення даних.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

Компоненти системи реалізовано з використанням наступних технологій і бібліотек:

- C# [1] - об'єктно-орієнтована мова програмування загального призначення. C# відноситься до сім'ї мов з C-подібним синтаксисом, їх синтаксис найбільш близький до C++ і Java. Мова має статичну типізацію, підтримує поліморфізм, перевантаження операторів (у тому числі операторів явного та неявного приведення типу), делегати, атрибути, події, змінні, властивості, узагальнені типи та методи, ітератори, анонімні функції з підтримкою замикань, LINQ, винятки, коментарі у форматі XML;
- Entity Framework — об'єктно-орієнтована технологія доступу до даних є object-relational mapping рішенням для .NET Framework від Microsoft. Надає можливість взаємодії з об'єктами як за допомогою LINQ як LINQ to Entities, так і з використанням Entity SQL;
- WPF (Windows Presentation Foundation) [10] - аналог WinForms, система для побудови клієнтських програм Windows з візуально привабливими можливостями взаємодії з користувачем, графічна підсистема у складі .NET Framework, що використовує мову XAML;
- JSON (JavaScript Object Notation) - простий формат обміну даними, зручний для читання та написання як людиною, так і комп'ютером.

JSON заснований на двох структурах даних:

- колекція пар ключ/значення. У різних мовах ця концепція реалізована як об'єкт, запис, структура, словник, хеш, іменованний список чи асоціативний масив;
- упорядкований перелік значень. У більшості мов це реалізовано як масив, вектор, список чи послідовність.

JSON - це універсальні структури даних. Майже всі сучасні мови програмування підтримують їх у будь-якій формі. Логічно припустити, що формат даних, незалежний від мови програмування, має ґрунтуватися на цих структурах.

SQLite - це реляційна база даних (RDBMS), що використовується для зберігання та управління структурованими даними. Вона є однією з найпопулярніших відкритих систем управління базами даних і використовується в широкому спектрі додатків, від невеликих додатків до великих корпоративних систем.

Основні характеристики SQLite [7]:

- реляційна структура: SQLite зберігає дані у вигляді таблиць зі стовпцями і рядками. Вона підтримує відносини між таблицями, що дозволяє ефективно управляти даними і забезпечує цілісність даних;
- мова запитів: SQLite використовує мову запитів SQL (Structured Query Language) для взаємодії з базою даних. SQL надає потужні можливості для створення, читання, оновлення та видалення даних;
- підтримка індексів: SQLite підтримує індекси, які допомагають прискорити пошук та фільтрацію даних. Індекси дозволяють ефективно виконувати запити і забезпечують швидкий доступ до даних;
- транзакції: SQLite підтримує транзакції, що забезпечують атомарність, консистентність, ізолюваність та стійкість (ACID properties) під час операцій з базою даних. Це важливо для забезпечення цілісності даних та уникнення проблем конкуренції при одночасному доступі до даних;
- масштабованість: SQLite дозволяє масштабувати базу даних, починаючи з невеликих проектів і розширюючись до великих розподілених систем. Вона підтримує горизонтальне масштабування (шарування) і вертикальне масштабування (покращення обладнання);
- безпека: SQLite надає можливості для захисту даних, включаючи аутентифікацію користувачів, контроль доступу та шифрування даних. Вона дозволяє налаштовувати рівні доступу для різних користувачів і забезпечує захист від несанкціонованого доступу до даних.

SQLite широко використовується як основна база даних для консольних додатків, систем управління вмістом, електронної комерції, аналітики даних та багатьох інших сфер. Вона має велику спільноту розробників і підтримується на

різних платформах, що робить її потужним інструментом для зберігання та управління даними.

4.1 Вивантаження габаритів та переліку приміщень з Autodesk Revit

Вивантаження габаритів та переліку приміщень з Autodesk Revit [3] можна здійснити за допомогою використання Revit API. В основі цього процесу лежить отримання доступу до елементів категорії OST_Rooms (приміщення) та зчитування необхідних даних.

Revit API – це набір бібліотек, що дозволяє взаємодіяти з моделями Revit програмно. За допомогою Revit API можна отримувати, змінювати та додавати елементи до моделі, а також автоматизувати різні процеси проектування.

OST_Rooms – це категорія елементів у Revit, яка представляє приміщення. Приміщення мають властивості, такі як назва, номер, площа, об'єм, висота та інші параметри, які можуть бути використані для аналізу та вивантаження даних.

Для отримання даних про приміщення необхідно створити фільтр, який відбиратиме елементи категорії OST_Rooms. Використовуючи цей фільтр, можна отримати колекцію приміщень та зчитувати їх властивості.

Основні властивості приміщень, які можуть бути корисними для вивантаження, включають:

- Name (назва приміщення);
- Number (номер приміщення);
- Area (площа приміщення);
- Volume (об'єм приміщення);
- Location (розташування приміщення);
- Perimeter (периметр приміщення).

Для цього необхідно імпортувати наступні бібліотеки:

- using Autodesk.Revit.DB;
- using Autodesk.Revit.UI;
- using System;
- using System.Collections.Generic;

- using System.Linq.

Надалі надано метод для вивантаження даних:

```

public Result Execute(ExternalCommandData commandData, ref string
message, ElementSet elements)
{
    UIApplication uiApp = commandData.Application;
    Document doc = uiApp.ActiveUIDocument.Document;
    // Отримання всіх приміщень у проекті
    FilteredElementCollector collector = new
FilteredElementCollector(doc);
    ICollection<Element> rooms =
collector.OfCategory(BuiltInCategory.OST_Rooms).WhereElementIsNotElementTyp
e().ToElements();
    // Створення списку для зберігання даних
    List<RoomData> roomDataList = new List<RoomData>();
    foreach (Element roomElement in rooms)
    {
        Room room = roomElement as Room;
        if (room != null)
        {
            string roomName = room.Name;
            string roomNumber = room.Number;
            double roomArea = room.Area;
            XYZ roomLocation = (room.Location as LocationPoint).Point;
            RoomData roomData = new RoomData
            {
                Name = roomName,
                Number = roomNumber,
                Area = roomArea,
                Location = roomLocation
            };
            roomDataList.Add(roomData);
        }
    }
    // Виведення або збереження даних (наприклад, у файл)
    SaveRoomDataToFile(roomDataList);
    return Result.Succeeded;
}

```

Вивантаження відбувається до бази даних SQLite [7].

4.2 Виконання вентиляційних розрахунків (у відповідності до [1], [2])

Вентиляційні розрахунки виконані у відповідності до ДБН [1], [2] та нормативної бази ASHRAE [9]. Системи вентиляції, кондиціонування повітря та опалення сховищ та СПП мають забезпечувати безперервне перебування у них людей, що підлягають укриттю впродовж 48 годин безперервно, зокрема для

сховищ з двома режимами вентиляції – впродовж 12 годин безперервно у режимі фільтровентиляції, а для сховищ з третім режимом вентиляції – впродовж 6 годин безперервно у режимі регенерації повітря.

Вентиляція сховищ та СПП здійснюється з використанням систем вентиляції з механічним спонуканням та/або із застосуванням електроручних вентиляторів.

В обов'язковому порядку має передбачатися очищення припливного повітря від твердих часток та пилу шляхом встановлення фільтрів грубого очищення повітря.

Систему вентиляції сховищ та СПП необхідно проєктувати на два режими: чистої вентиляції (режим I) та фільтровентиляції (режим II).

При режимі чистої вентиляції (режим I) подача у сховище та СПП очищеного від пилу зовнішнього повітря повинна забезпечувати необхідний обмін повітря та видалення з приміщень тепловиділень та вологи.

При режимі фільтровентиляції (режим II) зовнішнє повітря, яке подається у сховище та СПП, повинно очищатися від газоподібних засобів масового ураження, аерозолів та пилу, в тому числі від бойових отруйних, небезпечних хімічних і радіоактивних речовин та біологічних засобів, продуктів горіння.

Режим регенерації внутрішнього повітря (режим III) передбачається у сховищах та СПП, які розташовані у місцях можливої небезпечної загазованості повітря продуктами горіння, у зонах можливого хімічного забруднення небезпечними хімічними речовинами (НХР), а також у сховищах та СПП, які розташовані у зонах можливого затоплення. Необхідність передбачення режиму регенерації внутрішнього повітря (режиму III) зазначається в технічному завданні на проєктування.

Кількість зовнішнього повітря, яке подається у захисні споруди та СПП, слід приймати:

- при чистій вентиляції (режим I) – визначається за формулою 4.1, але не менше ніж значення, наведеного в таблиці 4.1;
- при фільтровентиляції (режим II) – визначається за формулою 4.2, але не менше, ніж: 2 м³/год на одну особу, яка підлягає укриттю; 5 м³/год на одного працюючого

у приміщеннях пункту керування; 10 м³/год на одного працюючого у фільтровентиляційній камері з електроручними вентиляторами.

Таблиця 4.1 – Нормативна мінімальна кількість повітря, яке подається в захисну споруду і СПП в режимі чистої вентиляції (режим I)

Кліматичні райони	Кількість повітря, яке подається, м ³ /люд·год
I, III, V	10
II, IV	11

Розрахункова кількість повітря, яке подається захисну споруду та СПП при режимі I (чиставентиляція), м³/год, визначається за формулою:

$$L = Q_T / (1,2 (I_{вн} - I_3)), \quad (4.1)$$

де Q_T – кількість тепла, яке виділяється від людей, електричного освітлення, електросилового обладнання, ккал/год;

I_3 – тепловміст зовнішнього повітря, який відповідає середньомісячній температурі та вологості найтеплішого місяця, ккал/кг;

$I_{вн}$ – тепловміст внутрішнього повітря, який відповідає допустимим сполученням температури та вологості повітря, ккал/кг (визначається відповідно до норм [1] у залежності від розрахункових тепловмісту I_3 та вологовмісту $I_{вн}$ зовнішнього повітря по I-d діаграмі відповідного кліматичного району) .

Кількість зовнішнього повітря у режимі фільтровентиляції (режим II) приймається з урахуванням вимог п.11.2.1.12 [1] та визначається за формулою:

$$L = (Q_T - F_k q_{огор}) / 1,2 (I_{вн} - I_3), \quad [м^3/год], \quad (4.2)$$

де $q_{огор}$ – кількість тепла, ккал/(год·м²), яка поглинається 1 м² огорожувальних конструкцій та приймається за таблицею 4.3;

F_k – площа внутрішньої поверхні зовнішніх огорожувальних конструкцій, м²;

$I_{\text{вн}}$ – тепловміст внутрішнього повітря, який приймається для I, III, V кліматичних районів) – 22,5 ккал/кг, для II, IV кліматичного району – 23,5 ккал/кг.

При цьому мінімальна кратність повітрообміну у приміщеннях для перебування осіб, які підлягають укриттю у I-му режимі (чиста вентиляція), має становити 6 за годину (4 при капремонті і реконструкції), для закладів охорони здоров'я - 10 за годину (6 при капремонті і реконструкції).

Кількість повітря, що подається до сховищ та СПП для закладів охорони здоров'я, а також для осіб, які працюють у медичних пунктах, приймається:

- при режимі I (чиста вентиляція) – за таблицею 4.2 з коефіцієнтом 1,5;
- при режимі II (фільтровентиляція) – з розрахунку 10 м³/год на одну особу, яка підлягає укриттю.

Кількість виділення людьми тепла та вологи слід приймати згідно з таблицею 4.2.

Таблиця 4.2 – Кількість тепла та вологи, що виділяються у сховищах і СПП

Показник	Кількість тепла та вологи, що виділяються у сховищах, розташованих	
	на підприємствах	при лікувальних закладах
Тепловиділення (повні) на одну особу, що підлягає укриттю	116 Вт (100 ккал/год)	116 Вт (100 ккал/год)
Вологовиділення, на одну особу, що підлягає укриттю, при температурі приміщень, °С:		
28	95 г/год	95 г/год
30	110 г/год	–
Тепловиділення (повні) від працюючого і обслуговуючого персоналу: хірурга, операційних сестер працюючих, що підлягають укриттю	–	203 Вт (75 ккал/год) 174 Вт (150 ккал/год)
Вологовиділення від персоналу: члени операційних бригад працівники закладу охорони здоров'я	–	200 г/год 170 г/год

Тепловиділення від електричного освітлення $Q_{\text{осв}}$, ккал/год (Вт) слід визначати за формулою 4.3:

$$Q_{\text{осв}} = 860 \times P_{\text{осв}}, [\text{ккал/год}] \text{ або } Q_{\text{осв}} = 1,16 \times 860 \times P_{\text{осв}}, [\text{Вт}], \quad (4.3)$$

де $P_{\text{осв}}$ – сумарна потужність джерел освітлення, кВт.

У випадку застосування газорозрядних ламп отримане за формулою 4.3 значення слід помножити на 0,15, при застосуванні світильників на світлодіодних лампах – помножити на 0,2.

Теплопоглинання цогор огорожувальними конструкціями повинно враховуватись лише для одного з режимів, як правило, для II-го режиму. У разі, якщо у технічному завданні на проектування сховища/СПП III режим вентиляції передбачається основним, то теплопоглинання враховується тільки для III режиму.

Теплопоглинання огорожувальними конструкціями сховищ враховується тільки за наявності обвалування.

Вентиляція приміщень, призначених для встановлення автономних джерел енергозабезпечення (електроживлення), а також чистої та забрудненої зон санпропускників, має передбачатися через окремі повітроводи від повітроводів інших систем вентиляції.

Повітрозабори мають розміщуватися поза межами можливих завалів будівель і споруд.

Повітрозабори систем фільтровентиляції допускається розміщувати на території зон можливих завалів та у передтамбурі захисної споруди (споруди подвійного призначення).

Повітрозабірний короб чистої вентиляції сховищ та СПП (режиму I) доцільно сполучати з аварійним виходом із сховища. У такому разі висота та розташування повітрозабірного коробу приймається з урахуванням вимог п. 7.5.1.19 [1] та 7.5.1.20 [1] цих норм.

Вимоги щодо мінімальної відстані між пристроєм для забору зовнішнього повітря та пристроєм для викиду витяжного повітря у залежності від якості витяжного повітря, яке видаляється системою механічної вентиляції, коефіцієнта розсіювання і витрати повітря через викидний пристрій, слід визначати відповідно до ДСТУ Б EN 13779.

У місцях розташування захисних споруд (споруд подвійного призначення) у житловій зоні населених пунктів допускається поєднання у загальних шахтах з

витяжних каналів із окремих приміщень захисних споруд та СПП та каналів вихлопних систем автономних джерел енергозабезпечення за умови використання повітропроводів зварними без рознімних з'єднань класу щільності не менше В згідно з ДБН В.2.5-67. Для цих повітропроводів слід передбачати окрему шахту з огорожувальними конструкціями класом вогнестійкості не менше EI120.

Кількість тепла, ккал/(год·м²), яка поглинається 1 м² огорожувальних конструкцій відображена у таблиці 4.3.

Таблиця 4.3 – Кількість тепла, яка поглинається огорожувальними конструкціями

Початкова температура огорожувальних конструкцій, °С	Середньогодинна кількість тепла, яка вбирається огорожувальними конструкціями, ккал/(год·м ²)					
	залізобетонними і бетонними			цегляною кладкою		
	при II-му режимі	при III-му режимі та температурі у приміщенні, °С		при II-му режимі	при III-му режимі та температурі у приміщенні, °С	
		32	31		32	31
15	92	139	129	56	85	80
16	85	129	120	52	80	74
17	78	120	110	48	74	68
18	72	110	101	44	68	62
19	65	101	91	39	62	56
20	58	91	81	35	56	50
21	50	81	72	31	50	44
22	43	72	62	27	44	38
23	36	62	53	22	38	32
24	30	53	43	18	32	27
25	24	43	34	14	27	21
26	16	34	24	10	21	15
27	9	24	14	2	15	9

Примітка 1. Початкова температура поверхні огорожувальних конструкцій приймається рівною середньомісячній температурі зовнішнього повітря найтеплішого місяця згідно з ДСТУ-Н Б В.1.1-27, але не нижче 15 °С.

Повітроводи припливних та витяжних систем, що прокладаються зовні, виконуються із будівельних конструкцій, розрахованих на дію повітряної ударної хвилі, або монтуються зі сталевих зварних труб і повинні прокладатися ухилом 0,003 і більше у бік від захисної споруди (споруди подвійного призначення). При цьому перед противибуховим пристроєм слід передбачати відведення конденсату.

Із сталевих труб з товщиною стінки не менше ніж 8 мм слід виготовляти повітроводи сховищ та СПП, що прокладаються всередині приміщень до герметичних клапанів, з'єднувальні повітроводи між повітрозаборами чистої вентиляції та фільтровентиляції, а також патрубки для установки герметичних клапанів у стінах.

Повітроводи фільтрів-поглиначів та регенеративних установок необхідно виготовляти зі сталеві труби або листові сталі завтовшки не менше ніж 2 мм.

Повітроводи всередині приміщення після герметичних клапанів та фільтрів виконують із сталевих труб круглого або прямокутного перерізу. Товщину листові сталі для повітроводів, уздовж яких переміщується повітря з температурою не вище ніж 80 0С, слід приймати згідно з ДБН В.2.5-67.

Довжина повітроводу від вентилятора до найбільш віддаленого вентиляційного отвору має бути:

- для систем вентиляції з електроручними вентиляторами – не більше ніж 30 м;
- для систем, обладнаних промисловими вентиляторами з електроприводами рекомендується не більше ніж 50 м.

Повітроводи, по яких транспортується повітря з високою температурою, мають бути гідроізольованими та теплоізольованими.

Для забезпечення експлуатаційного підпору 50 Па (5 кгс/м²) при II-му режимі кількість припливного повітря у сховище/СПП L_{II} (м³/год) має бути не менше суми величин, щокомпенсують витікання крізь огорожувальні конструкції, витяжку з санвузлів, станції перекачки дренажних вод (за її наявності), а також перетікання повітря із приміщень сховища/СПП у приміщення ДЕС (при вентиляції ДЕС повітрям із основних приміщень сховища/СПП).

Кількість повітря L_{II}, (м³/год), для забезпечення експлуатаційного підпору 50 Па (5 кгс/м²) при II-му режимі слід визначати за формулою:

$$L_{II} \geq K_{II} F_{огор} + L_{св} + L_{сп} + L_{дес} + L_{сп}', \quad (4.4)$$

де K_{II} - питоме витікання (втрата) повітря, м³/год, через 1 м² огорожувальних конструкцій по контуру герметизації сховища і приймається: для сховищ звичайної герметичності – 0,53 м³/м²·год, для сховищ підвищеної герметичності – 0,22 м³/м²·год;

$F_{огор}$ – площа огорожувальних конструкцій сховища по контуру герметизації, м²;

$L_{св}$ – кількість повітря, яке видаляється з санвузлів, м³/год;

$L_{сп}$ – кількість повітря, яке видаляється із станції перекачування дренажних вод, м³/год;

$L_{дес}$ – кількість повітря, яке надходить у приміщення ДЕС з приміщень для осіб, які підлягають укриттю, при II режимі, м³/год;

$L_{сп'}$ – повна кількість повітря, яке видаляється з чистої та забрудненої зон для сховищ/СПП, які мають санпропускники, приміщення для запасу води, комору для зберігання продуктів тощо.

У сховищах, обладнаних санпропускником, кількість повітря, яке подається у сховище/СПП, зростає на величину, яка визначається за формулою:

$$L'_{сп} = \frac{Q'_{сп}}{\gamma' \cdot c \cdot (t_n - t_3)}, \quad (4.5)$$

де $Q_{сп'}$ – повна кількість тепла, яке виділяється у сховище/СПП, зокрема, що виділяється при пароутворенні у системах водопостачання (душовими, бойлерами тощо), ккал/год;

c – теплоємність повітря, яка приймається 0,24 ккал/кг·°С, (1,028 кДж/кг·К);

γ' – об'ємна вага повітря, яка приймається 1,2 кгс/м³;

t_n – температура повітря у приміщенні душової, яка приймається 25 °С;

t_3 – розрахункова середньомісячна температура зовнішнього повітря найтеплішого місяця.

$$Q'_{сп} = (t_2 - t_1)(c_1 m_n n + c_2 m_e), \quad (4.6)$$

де t_2 – температура нагрітої води електроводонагрівачами, яка приймається 80 °С;

t_1 – початкова температура санпропускника (температура 1 повітря всередині санпропускника), яка приймається 16 град. °С;

c_1 – питома теплоємність матеріалу електронагрівача, яка приймається як для заліза,

ккал/кг·°С;

m_H – маса електронагрівача;

n – кількість електронагрівачів;

m_B – маса запасу води для миття людей у санпропускнику сховища;

c_2 – питома теплоємність води, ккал/кг·°С.

У сховищах атомних енергетичних об'єктів, що мають санпропускник, повітрообмін приймається:

- в душовій приплив 10-кратний, витяжка 11-кратна за годину незалежно від режиму вентиляції;
- у роздягальні, приміщенні для брудного одягу, у приміщенні для дозконтролю приплив 2,5-кратний, витяжка 2-кратна;
- у роздягальні, приміщенні для чистого одягу приплив 2-кратний, витяжка 1,5-кратна.

У вікні додатку відображеному на рисунках 4.1, 4.2 отримаємо розрахункову інформацію по кожному приміщенню, системі, зоні.

Суммарна кількість припливного повітря по будівлі: 7577, м³/год
Суммарна кількість витяжного повітря по будівлі: 7577, м³/год
Різниця складає: 0, м³/год

Зона вентиляції: MAgea3

Кількість припливного повітря : 518 м³/год
Кількість витяжного повітря : 333 м³/год
Різниця : 185 м³/год

Зона вентиляції: MAgea4

Кількість припливного повітря : 7059 м³/год
Кількість витяжного повітря : 7244 м³/год
Різниця : -185 м³/год

Кількість припливного повітря по системі: П1
Становить : 185 м³/год

Кількість припливного повітря по системі: П3
Становить : 333 м³/год

Кількість припливного повітря по системі: П2
Становить : 0 м³/год

Кількість припливного повітря по системі: П12
Становить : 60 м³/год

Кількість припливного повітря по системі: П13
Становить : 6999 м³/год

Рисунок 4.1 – Виведення розрахунків

```

=====
#1_Розрахункове приміщення
=====
    Тип системи: Припливно-витяжна механічна
Номер приміщення: 101
Назва приміщення: Сходова клітини Н4 (2)
Температура приміщення, град. С: 5
Розрахунковий рівень приміщення: Level-1_Test
Площа, м2: 22.2
Висота повна, м: 3.5
Висота розрахункова, м: 3.5
Зона вентиляції: MArea3
=====

-----
    Розрахунок за кратністю |
-----

Розрахункова кількість припливного повітря, м3/год: 333
Назва системи (приплив): ПЗ
Кратність повітрообміну у приміщенні (приплив), об./год.: 5
Загальна формула:
Кількість повітря = (Площа * Висота розрахункова) * Кратність повітря, [м3/год]
Приплив повітря за формулою:
333 = (22.2 * 3.5) * 5, м3/год
Наявність перетоку [прим.-прим.]: 101->104
Номер переточного приміщення: 104

Розрахункова кількість витяжного повітря, м3/час: 333
Назва системи (витяжка): В12
Кратність повітрообміну у приміщенні (витяжка), об./год.: 5
Витяжка повітря за формулою:
333 = (22.2 * 3.5) * 5, м3/год
Наявність перетоку [прим.-прим.]: -
Номер переточного приміщення: 101

```

Рисунок 4.2 – Загальний вид програми з вихідними даними

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування програмного забезпечення є важливим етапом розробки, який дозволяє виявити та виправити помилки, забезпечити якість та надійність продукту. У цьому розділі буде розглянуто підходи до тестування програмного забезпечення для розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів. Особлива увага буде приділена юніт-тестуванню, інтеграційному тестуванню та тестуванню користувацького інтерфейсу.

На цьому етапі проводиться тестування програмного забезпечення для перевірки відповідності вимогам та виявлення помилок. Це включає функціональне тестування для перевірки правильності роботи програми та нефункціональне тестування для перевірки продуктивності, безпеки та стійкості програми. Після виявлення помилок вони виправляються, а програма перевіряється знову. Процес тестування та відлагодження повторюється до досягнення необхідного рівня якості програмного забезпечення.

Під час розробки програмного забезпечення для розрахунку систем вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів використовуються різні методи тестування для перевірки правильності та надійності роботи програми.

Приклад юніт-тесту для методу розрахунку об'єму приміщення:

```
using NUnit.Framework;
using VentilationApp.Models;

namespace VentilationApp.Tests
{
    [TestFixture]
    public class RoomTests
    {
        [Test]
        public void CalculateVolume_CorrectValues_ReturnsCorrectVolume ()
        {
            // Arrange
            var room = new Room
            {
                Length = 10,
                Width = 5,
                Height = 3
            };
        }
    }
}
```

```

        // Act
        var volume = room.Volume;
        // Assert
        Assert.AreEqual(150, volume);
    }
}
}

```

У цьому тесті ми перевіряємо, що метод `Volume` класу `Room` правильно обчислює об'єм приміщення на основі заданих розмірів. Такий тест допомагає впевнитися в коректності роботи методу при різних вхідних даних.

Інтеграційне тестування (*integration testing*) – це метод тестування, який перевіряє взаємодію між окремими компонентами або модулями програми. Метою інтеграційного тестування є виявлення помилок у взаємодії між компонентами.

Приклад інтеграційного тесту:

```

using NUnit.Framework;
using VentilationApp.Models;
using VentilationApp.Services;

namespace VentilationApp.Tests
{
    [TestFixture]
    public class VentilationSystemIntegrationTests
    {
        private DataRepository _repository;
        [SetUp]
        public void Setup()
        {
            _repository = new DataRepository("TestDatabase.db");
        }
        [Test]
        public void AddVentilationSystem_ValidData_SavesCorrectly()
        {
            // Arrange
            var system = new VentilationSystem
            {
                Name = "Test System",
                Rooms = new List<Room>
                {
                    new Room { Length = 10, Width = 5, Height = 3 }
                },
                VentilationUnits = new List<VentilationUnit>
                {
                    new VentilationUnit { Type = "Приточна", Capacity = 500
}
                }
            };

            // Act
            _repository.AddVentilationSystem(system);

```

```

        var savedSystem =
_repository.GetVentilationSystem(system.Id);

        // Assert
Assert.NotNull(savedSystem);
Assert.AreEqual("Test System", savedSystem.Name);
Assert.AreEqual(1, savedSystem.Rooms.Count);
Assert.AreEqual(1, savedSystem.VentilationUnits.Count);
    }
}
}

```

Цей тест перевіряє, що вентиляційна система правильно зберігається у базі даних, а її компоненти (приміщення та вентиляційні установки) зберігаються та зв'язуються коректно.

Тестування користувацького інтерфейсу (UI testing) – це метод тестування, який перевіряє правильність роботи інтерфейсу користувача. Метою UI тестування є виявлення помилок у роботі інтерфейсу та перевірка його зручності для користувачів.

Тест-план включає перелік функціональностей, які необхідно протестувати, методи тестування та критерії прийняття. Нижче наведено приклади тест-кейсів для основних функцій програми.

Тест-кейс 1: Розрахунок об'єму приміщення

Назва: Розрахунок об'єму приміщення

Опис: Перевірка коректності розрахунку об'єму приміщення за заданими розмірами.

Передумови: Приміщення з довжиною 10 м, шириною 5 м та висотою 3 м.

Кроки тесту:

- відкрити форму введення даних;
- ввести довжину 10 м;
- ввести ширину 5 м;
- ввести висоту 3 м;
- натиснути кнопку "Розрахувати".

Очікуваний результат: Об'єм приміщення становить 150 кубічних метрів.

Тест-кейс 2: Додавання вентиляційної системи

Назва: Додавання вентиляційної системи

Опис: Перевірка коректності додавання нової вентиляційної системи з приміщенням та вентиляційними установками.

Передумови: Відсутність вентиляційних систем у базі даних.

Кроки:

- відкрити форму введення даних;
- ввести назву системи "Test System";
- додати приміщення з розмірами 10 м x 5 м x 3 м;
- додати вентиляційну установку типу "Приточна" з потужністю 500 м³/год;
- натиснути кнопку "Зберегти".

Очікуваний результат: Вентиляційна система збережена в базі даних з правильними даними.

ВИСНОВКИ

Результатом виконання роботи є програмна система для здійснення можливості інженерних розрахунків.

Застосунок створений за допомогою мови програмування C#, дані зберігаються в системі управління базами даних SQLite.

Під час проведеного аналізу предметної галузі вентиляції підземних споруд подвійного призначення та протирадіаційних укриттів було виявлено декілька основних проблем та викликів:

- недоліки існуючих програм для розрахунку вентиляції підземних споруд. Багато з наявних комерційних програм мають обмежений функціонал або не враховують специфічні вимоги та особливості підземних споруд.
- необхідність інноваційних підходів. У галузі вентиляції підземних споруд існує потреба в розвитку нових технологій та інноваційних підходів, які б забезпечували ефективність та безпеку систем вентиляції.
- необхідність покращення якості програмного забезпечення. Багато існуючих програм потребують вдосконалення у відношенні надійності, продуктивності та можливостей адаптації до різних умов експлуатації.
- потреба у зручних та інтуїтивних інструментах. Користувачі вимагають зручних та інтуїтивних інструментів для розрахунку та аналізу систем вентиляції, що дозволить їм ефективно виконувати свої завдання.

В цілому, перспективи для подальшого розвитку програмного забезпечення для розрахунку систем вентиляції підземних споруд є значними, оскільки в цій галузі постійно виникають нові вимоги та виклики, на які програмне забезпечення може відповідати шляхом постійного вдосконалення своєї функціональності та ефективності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДБН В.2.2-5:2023 Захисні споруди цивільного призначення.
2. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування.
3. Autodesk Revit: BIM software to design and make anything. URL: www.autodesk.com/products/revit/overview?term=1-YEAR&tab=subscription&plc=RVT (дата звернення 10.06.2024).
4. Carrier: Software for system design engineers by system design engineers. URL: www.carrier.com/commercial/en/us/software/hvac-system-design (дата звернення: 10.07.2024).
5. ССК ТМ: Вентиляція для бомбосховищ. URL: www.ccktm.com/cardfile/ (дата звернення: 11.07.2024).
6. MS Excel: Електронні таблиці. URL: <https://www.microsoft.com/uk-ua/microsoft-365/excel/spreadsheets> (дата звернення: 11.06.2024).
7. SQLite: Полегшена реляційна база даних. URL: SQLite.org. (дата звернення: 18.06.2024).
8. Troelsen, A., & Japikse, P. (2017). Pro C# 7: With .NET and .NET Core. Apress.
9. ASHRAE Handbook—Fundamentals. (2021). American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.
10. Sharp, J., & Nagel, C. (2019). Windows Presentation Foundation Development Cookbook. Packt Publishing.

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

StrikePlagiarism.com  Дата звігу 7/17/2024
Дата редагування ---  Звіт не був оцінений.

метадані





Заголовок
2024_2_ПІ_Пр_ПЗПп_22_1_ТельнийВМ_диплом_17_07_плагіат

Автор Науковий керівник / Експерт
Тельний Віталій Миколайович Вадим Юрійович Нечволод

підрозділ
Харківський національний університет радіоелектроніки

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		50
Інтервали		1
Мікропробіли		10
Білі знаки		0
Парафрази (SmartMarks)		72

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2



10773

Кількість слів

88519

Кількість символів

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз		Копір тексту	
ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	https://e-construction.gov.ua/laws_detail/3225773063500990463?doc_type=2	405	3.76 %
2	2024_Б_ПІ_ПЗПп_22_2_Душко_А_В_скорочений 7/11/2024 Kharkiv National University of Radio Electronics (Харківський національний університет радіоелектроніки)	227	2.11 %
3	https://e-construction.gov.ua/laws_detail/3225773063500990463?doc_type=2	206	1.91 %
4	https://e-construction.gov.ua/laws_detail/3225773063500990463?doc_type=2	200	1.86 %
5	https://e-construction.gov.ua/laws_detail/3225773063500990463?doc_type=2	141	1.31 %

ДОДАТОК Б

Слайди презентації

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кваліфікаційна робота бакалавра

1

ПРОГРАМА ДЛЯ РОЗРАХУНКУ СИСТЕМ ВЕНТИЛЯЦІЇ ПІДЗЕМНИХ СПОРУД ПОДВІЙНОГО ПРИЗНАЧЕННЯ ТА ПРУ

Виконав:
студент гр. ПЗПп-22-1
Тельний В. М.

Науковий керівник:
к.т.н., доцент каф. ПІ
Колесников Д.О.

2

► Мета роботи

- Автоматизація розрахунків вентиляційних систем
- облік розрахунків та вихідних даних
- розподіл розрахунків між приміщеннями
- облік припливного та витяжного повітря систем будівлі
- аналіз даних та баланс повітрообміну зон вентиляції
- інтеграція з програмою Autodesk Revit

Принципова схема

3



► Аналіз аналогів

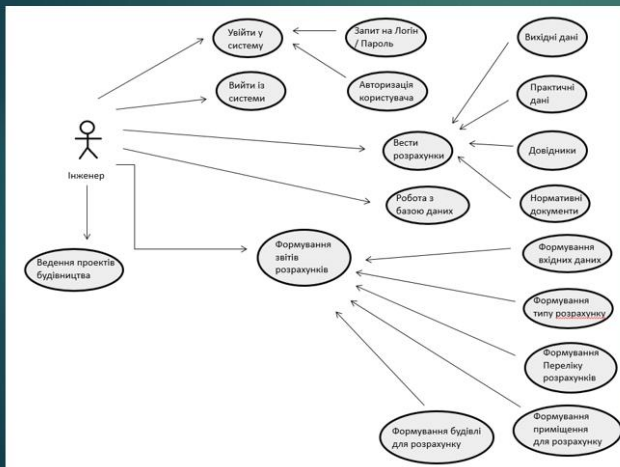
4

Рішення	Програма <u>css duct</u> ТМ ССК, Харків	Різноманітні інженерні програми на базі MS Excel	Carrier HAP
<u>Переваги</u>	Простота використання. Точний підбір вентиляційного обладнання компанії ССК. База даних: Містить бібліотеки з характеристиками обладнання компанії ССК.	Простота використання. Точні розрахунки: Програмне забезпечення забезпечує точні розрахунки для проектування HVAC систем.	Простота використання: Інтерфейс користувача відносно простий та зручний для інженерів. Точні розрахунки: Програмне забезпечення забезпечує точні розрахунки для проектування HVAC систем. Велика база даних: Містить бібліотеки з характеристиками обладнання, що полегшує проектування.
<u>Недоліки</u>	<u>Відсутність бази даних обладнання інших компаній</u> Ізольованість: Неможлива інтеграція з іншими BIM-платформами.	<u>Відсутність бази даних обладнання</u> <u>Функціональність обмежена</u> MS Excel Ізольованість: Неможлива інтеграція з іншими BIM-платформами.	Обмежена функціональність: Менше функцій порівняно з більш комплексними системами. Ізольованість: Обмежені можливості інтеграції з іншими BIM-платформами. <u>Малofункціональний</u> інтерфейс: Деякі користувачі вважають інтерфейс застарілим.
<u>Вартість</u>	Безкоштовна	Безкоштовна	Безкоштовна

► Постановка задачі та опис системи

5

Спроекувати та реалізувати програмну для розрахунку систем вентиляції підземних споруд подвійного призначення та ПРУ

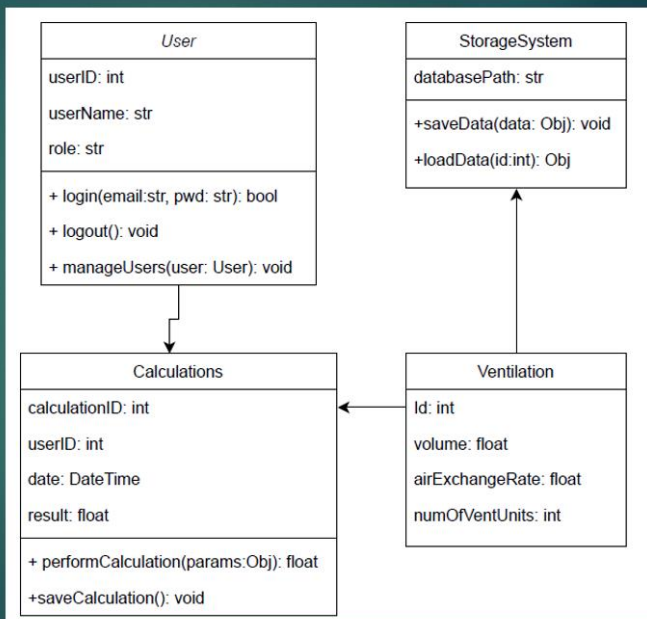


Інженер відповідає за наступне:

- Введення вихідних даних
- Вибір типу розрахунків для кожного приміщення
- Аналіз розрахункової інформації
- Повторні перерахунки у випадку потреби
- Формування та вивід розрахункової інформації в цілому по будівлі та індивідуально за кожним приміщенням

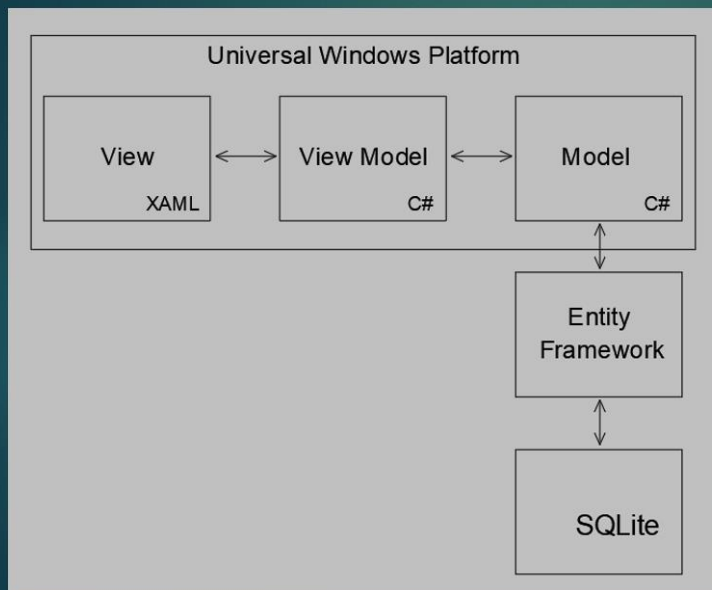
► Діаграма класів

6



Структурна схема

7



Вибір технологій розробки

мова програмування: C#

- система управління базами даних: SQLite, Entity Framework
- фреймворк для інтерфейсу користувача: WPF

► Логічна модель

8

ER -діаграма

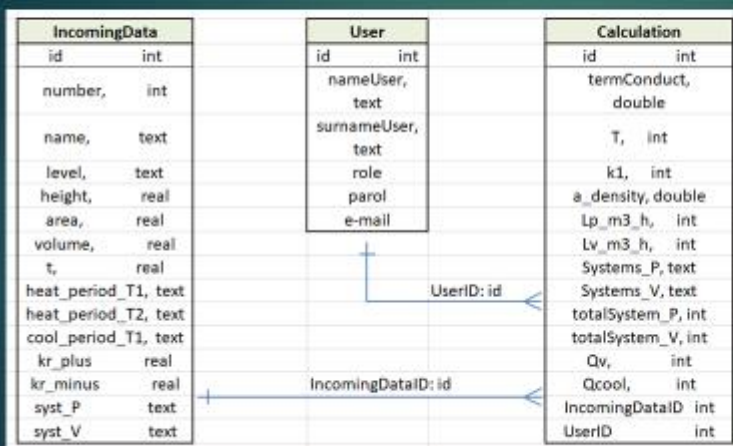
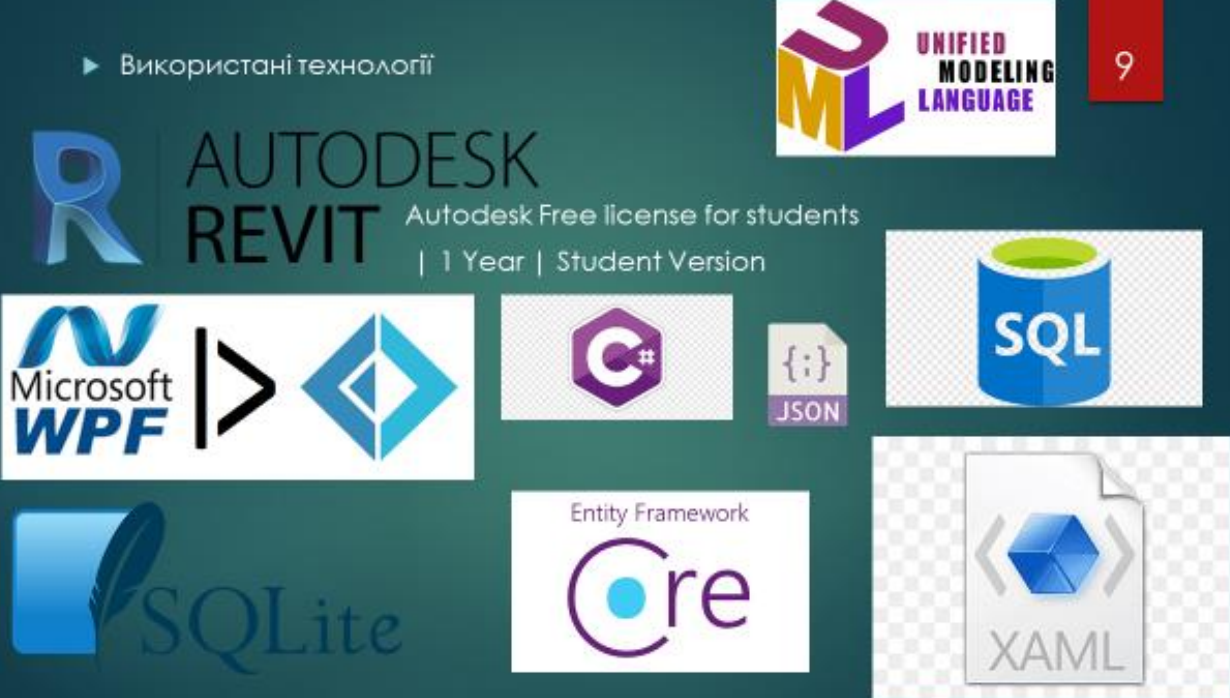


Схема обробки запитів



► Використані технології



AUTODESK REVIT Autodesk Free license for students
| 1 Year | Student Version

UNIFIED MODELING LANGUAGE

Microsoft WPF

SQL

SQLite

Entity Framework Core

JSON

XAML

► Приклад програмної реалізації розрахунку №1 «за кратністю приміщення»

```
public double Lkr_minus(double b_area_d, string H_str, string b_kr_minus) // розрахунок за кратністю (приплив)
{
    double Lkr_minus = (b_area_d * Convert.ToDouble(H_str) * Convert.ToDouble(b_kr_minus))/1000;
    Log($"{\n - kalk1.calculation.Lkr_minus" +
        $"{\nLkr_minus {Lkr_minus}");
    return Lkr_minus;
}

public double Lkr_plus(double b_area_d, string H_str, string b_kr_plus) // розрахунок за кратністю (витяжка)
{
    double Lkr_plus = (b_area_d * Convert.ToInt32(H_str) * Convert.ToInt32(b_kr_plus))/1000;
    Log($"{\n - kalk1.calculation.Lkr_plus" +
        $"{\nLkr_plus {Lkr_plus}");
    return Lkr_plus;
}
```

► Інтерфейс користувача

11

The screenshot shows a web application interface. On the left, there is a login form with fields for 'ІМ' (Name), 'Пароль' (Password), and 'Варіант' (Variant). Below the form are buttons for 'Приміщення' (Room) and 'Планш 3D'. On the right, there is a table titled 'Види результатів' (Results types) with columns: ID, Назва (Name), and several numerical columns. The table contains 24 rows of data.

ID	Назва	П2	П3	П4	П5	П6	П7	П8	П9	П10	П11	П12	П13	П14	П15	П16	П17	П18	П19	П20	П21	П22	П23	П24
101	Склад будівель №1 (2)	22,2	24,163	2,2	3	150	1	3	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
102	Панель-панель 1 на ступі	3,8	33,308	2	3,8	35	0	20	8	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
103	Забудова для життєвого об'єкта	2419	22,1186,2	2	2	220	585	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220
104	Матеріал будівельного матеріалу на 170см	12	33,43,1	1	1	119	19	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120	120
105	Висота	20,8	22,107,2	2	2	240	176	221	221	221	221	221	221	221	221	221	221	221	221	221	221	221	221	221
106	Склад будівель №1 (1)	22,2	24,163	2	2	240	22	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220
107	Панель-панель 1 на ступі	19,8	33,308	2	3,8	35	0	20	8	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
108	Оптимізація складової частини матеріалу будівельного	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
109	Оптимізація складової частини матеріалу будівельного	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
110	Склад будівель №1 (2)	22,2	24,163	2	2	240	22	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220
111	Склад будівель №1 (1)	22,2	24,163	2	2	240	22	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220
112	Приміщення в плані об'єкта	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
113	Склад будівель №1 (2)	22,2	24,163	2	2	240	22	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220
114	Склад будівель №1 (1)	22,2	24,163	2	2	240	22	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220
115	Приміщення в плані об'єкта	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
116	Приміщення в плані об'єкта	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
117	Приміщення в плані об'єкта	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
118	Приміщення в плані об'єкта	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
119	Приміщення в плані об'єкта	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
120	Приміщення в плані об'єкта	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
121	Приміщення в плані об'єкта	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
122	Приміщення в плані об'єкта	3,8	33,308	2	2	22	10	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
123	Матеріал будівельного матеріалу на 170см	19,8	33,308	2	2	220	58	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220	220
124	Панель-панель 1 на ступі	3,8	33,308	2	3,8	35	0	20	8	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

► Тестування

12

Тестовий випадок	Опис	Вхідні дані	Очікуваний результат	Фактичний результат	Статус
TC01: Завантаження координат приміщень будівлі №1	Завантаження координат приміщень будівлі №1	Данні про роботу: Назва, Номер, Площа, Висота, ОБ'єм, Координати XYZmin, XYZmax	Завантаження успішно створено, відображається у головному вікні програми	Завантаження успішно створено, відображається у головному вікні програми	Виконано
TC02: Вибір нормативної бази для розрахунків	Перевірка читання існуючого запису з БД	Ідентифікатор запису	Відображення всіх даних відповідного запису	Дані запису відображені	Виконано
TC03: Виконання розрахунку №1	Перевірка оновлення даних розрахунку в базі даних	Ідентифікатор запису, Нові дані	Запис успішно оновлено, зміни збережені	Запис оновлено	Виконано
TC04: Виведення інформації за приміщенням №105	Перевірка відображення інформації про приміщення	Ідентифікатор запису	Запис успішно надано у log.txt	Запис оновлено	Виконано

► Практична користь

13

- Ефективний розрахунок вентиляційних систем
- Оптимізація процесів отримання координат приміщень з 3д моделі будівлі
- Покращення процесу проектування вентиляційних систем будівлі
- Зменшення часу вентиляційних розрахунків
- Підвищення точності розрахунків

► Висновки

14

- Проведений аналіз предметної галузі
- Спроекована архітектура програмної системи
- Реалізована та протестована програмна система
- Закладені основи розвитку програмної системи для розрахунків вентиляційних систем



ДОДАТОК В

Приклади тест кейсів

Цей тест-кейс повинен бути виконаний для тестування функціонування результатів розрахунків до системи.

Інформація про тестовий випадок			
Ідентифікатор тестового випадку	SC04 ver1.0		
Власник тесту			
Місцезнаходження тесту(шлях)	TestServer:D:\Project\TestCases\SC04.doc		
Дата останнього перегляду	13/07/2024		
Технічна вимога, щотестується	SC102		
Конфігурація засобівтестування	ST02		
Взаємозалежність тестовихвипадків	Виконати прогін тесту SC04 і його налаштуванняперед прогоном даного тесту.		
Мета тесту	Перевірити, що допустимі вхідні значення температури у поле вводу при заповненні інформації для розрахунків і що недопустимі вхідні значення призводять до повідомлення про помилку.		
Методика тестування			
Налаштування прогону тесту	Не проводиться		N/A *
Крок	Дія	Очікуваний результат	Відмітка (V)*
1.	Не ввести нічого у поле Temp та натиснути кнопку Submit	Повідомлення про помилку „Temp is required”, будуть помилки в інших полях	(V)
2.	Поставити курсор на поле Temp.	Поле стало у фокусі	(V)
3.	Ввести „01” в поле Temp та натиснути кнопку Submit	Повідомлення про помилку „Temp can not contain number”, будуть помилки в інших полях	(V)
4.	Ввести „a” в поле Temp та натиснути	Повідомлення про помилку „Temp can not contain letters”	(V)

	кнопку Submit		
5.	Ввести „№:?*” в поле Temp та натиснути кнопку Submit	Повідомлення про помилку „Temp must contain only US characters”	(V)

Цей тест-кейс повинен бути виконаний для тестування функції пошуку приміщення (пошук, фільтрація) і для тестування функції відправки нотифікацій користувачам до системи.

Інформація про тестовий випадок			
Ідентифікатор тестового випадку	SC23 ver1.0		
Власник тесту			
Місцезнаходження тесту(шлях)	TestServer: D:\Project\TestCases\SC23.doc		
Дата останнього перегляду	13/07/2024		
Технічна вимога, що тестується	SC123		
Конфігурація засобів тестування	ST11		
Взаємозалежність тестових випадків	Виконати прогін тесту SC23 і його налаштування перед прогоном даного тесту.		
Мета тесту	Перевірити, що користувач може бачити інформацію про надіслані нотифікації, а також виконує пошук розрахунків та сортування.		
Методика тестування			
Налаштування прогону тесту	Не проводиться		N/A *
Крок	Дія	Очікуваний результат	Відмітка(V)*
1.	Перейти на сторінку зі списком надісланих заявок на донорство	На сторінці відображені заявки на донорство	(V)
2.	Не ввести у поле пошуку нічого та натиснути кнопку «Find».	Відображені всі заявки, як і були	(V)

3.	Ввести у поле пошуку прізвище «Smith» та натиснути кнопку «Find».	Показані заявки на донорство з полем Імя, яке у своєму складі має «Smith» без урахування реєстру або повідомлення про то, ще відповідні результати не знайдені.	(V)
4.	Натиснути «results» кнопку «Clear»	Результати очищені пошуку	(V)
5.	Ввести у поле пошуку №заяви «1354897» та натиснути кнопку «Find».	Показані повідомлення з полем Request number, яке у своєму складі має «1354897» або повідомлення про то, ще відповідні результати не знайдені.	(V)
6.	Натиснути на таблиці кнопку сортування за прізвищем у порядку зростання	Результати подані, відсортовані по прізвищу від А до Я	(V)
7.	Натиснути «results» кнопку «Clear»	Результати очищені сортування	(V)
8.	Натиснути на таблиці кнопку сортування за датою повідомлення у порядку спадання	Результати подані, відсортовані за датою від сьогодні до минулого.	(V)
9.	Натиснути на таблиці кнопку сортування за номером заяви у порядку спадання.	Результати подані, відсортовані за датою від сьогодні до минулого, а заяви, які мають однакову дату, відсортовані за номером від більшого до меншого.	(V)
Очистка після прогону теста	Очистити всі введені критерії пошуку та фільтри, вилогінітися з системи		(V)
Результати тесту			
Тестувальник: Тельний В.		Дата прогону тесту: 13/07/2024	Результат тесту (P/F/V)*: ПРОЙДЕНО (P)