

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»
Факультет автоматизації виробництва, інформаційних
та управлінських технологій
Кафедра інформаційних технологій та аналітики даних

«Допущено до захисту»
Гарант ОПП

Ірина ГЕТЬМАН

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

за підсумками виконання
освітньо-професійної програми
«Комп'ютерні науки»
за спеціальністю 122 Комп'ютерні науки

**на тему «Програмно методичний комплекс для інформаційної
підтримки процесу підбору сонячних панелей та інверторів
для альтернативного живленні домогосподарств України»**

Керівник роботи

Ірина ГЕТЬМАН

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають
посилання на відповідне джерело*

Здобувач

Сергій ТИЩЕНКО

Підсумкова оцінка за атестацію			
--------------------------------	--	--	--

Голова ЕК

Антон КУДРЯВЦЕВ

ЗАПОРІЖЖЯ 2026

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»

Факультет	автоматизації виробництва , інформаційних та управлінських технологій
Кафедра	інформаційних технологій та аналітики даних
Ступінь вищої освіти	бакалавр
Спеціальність	122 Комп'ютерні науки
ОПП	Комп'ютерні науки

ЗАТВЕРДЖУЮ

Гарант ОПП

_____ Ірина ГЕТЬМАН

«26» лютого 2026 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Тищенко Сергію Анатолійовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема роботи «Програмно методичний комплекс для інформаційної підтримки процесу підбору сонячних панелей та інверторів для альтернативного живленні домогосподарств України»

керівник роботи Гетьман Ірина Анатоліївна, доцент, канд. техн. наук

_____ (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом Університету від 23.02.2026 р. №41/23.02.2026

2. Термін подання роботи 16.06.2026 р.

3. Вихідні дані до роботи Навчальна література, державні стандарти та методичні вказівки з дисциплін «Розподілені обчислення та хмарні технології», «Алгоритми та структури даних», «Бази даних», «Стандарти та методології бізнес-аналізу», документація з розробки ПЗ на Javascript і платформі Node.js, наукові публікації з тематики альтернативної енергетики, дослідження в галузі побудови інформаційних систем, результати власного проектування, тестування, експериментальних розрахунків та економічного обґрунтування впровадження систем автономного живлення будинків.

4. Зміст пояснювальної записки (перелік питань) Реферат. Зміст. Вступ. Розділ 1. Аналіз предметної області та принципів моделювання системи підбору компонентів сонячної енергетики. Розділ 2. Математичне моделювання процесу підбору компонентів сонячної енергетики. Розділ 3. Проектування та реалізація програмного комплексу інформаційної підтримки процесу підбору сонячних панелей та інверторів для альтернативного живленні домогосподарств України . Розділ 4 Аналіз економічної доцільності розробки програмного комплексу для інформаційної підтримки процесу підбору сонячних панелей та інверторів для альтернативного живленні домогосподарств України. Висновки. Перелік використаних джерел. Додатки.

5. Перелік графічного (демонстраційного) матеріалу (з точним зазначенням обов'язкових креслень): Актуальність, мета, об'єкт, предмет та завдання дослідження; логічна структура альтернативних енергосистем; SADT-діаграми; ER-діаграма структури бази даних; діаграми UML (прецедентів, класів, послідовностей); інтерфейси форм користувача; приклади реалізованих звітів і зв'язків таблиць у СУБД; результати розрахунків витрат та терміну окупності системи; висновки до роботи; (за наявності) – копії публікацій або розробленої документації.

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх.

Розділ	Прізвище, ініціали та посада консультанта
1	Гетьман І.А., доц. каф. ЦТПАР
2	Гетьман І.А., доц. каф. ЦТПАР
3	Гетьман І.А., доц. каф. ЦТПАР
4	Гетьман І.А., доц. каф. ЦТПАР

7. Дата видачі завдання 26.02.2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи
1	Аналіз предметної області та принципів моделювання системи підбору компонентів сонячної енергетики.	26.02.2026 - 08.03.2026
2	Математичне моделювання процесу підбору компонентів сонячної енергетики	09.03.2026 - 23.03.2026
3	Проектування та реалізація програмного комплексу для інформаційної підтримки процесу підбору сонячних панелей та інверторів для альтернативного живленні домогосподарств України	24.03.2026 – 12.05.2026
5	Аналіз економічної доцільності розробки програмного комплексу для інформаційної підтримки процесу підбору сонячних панелей та інверторів для альтернативного живленні домогосподарств України	13.05.2026 - 30.05.2026
6	Висновки, перелік посилань, вступ, зміст, реферат	01.06.2026 – 12.06.2026
7	Остаточне оформлення та подання завершеної роботи.	08.02.2026 – 16.06.2026
8	Підготовка презентаційного матеріалу, рецензування завершеної роботи. Захист	16.06.2026 – 18.06.2026

Здобувач

(Сергій ТИЩЕНКО)

Керівник роботи

(Ірина ГЕТЬМАН)

РЕФЕРАТ

Тищенко С.А Програмно-методичний комплекс для інформаційної підтримки процесу підбору сонячних панелей та інверторів для альтернативного живлення домогосподарств України.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра комп'ютерних наук за спеціальністю 122 «Комп'ютерні науки». – ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» МОН України, м. Запоріжжя, 2026.

Метою роботи було створення програмно-методичного комплексу, що забезпечує інформаційну підтримку та автоматизацію процесу підбору сонячних панелей та інверторів для альтернативного живлення приватних домогосподарств України. Об'єктом дослідження є процес проєктування та підбору компонентів побутових фотоелектричних систем.

Предметом дослідження є методи та програмні засоби інформаційної підтримки технічного узгодження фотоелектричних модулів із силовими перетворювачами на основі моделювання електричних та енергетичних параметрів.

Методологія – системний аналіз предметної області, математичне моделювання енергетичного балансу, об'єктно-орієнтоване проєктування, методи перевірки електричної сумісності, алгоритмізація підбору конфігурацій, UML/SADT- моделювання інформаційних потоків.

У роботі проаналізовано стан ринку побутової сонячної енергетики, визначено сценарій розвитку предметної області та ключові поняття для моделювання. Розроблено математичну модель перевірки сумісності обладнання за технічними критеріями. Проведено кількісний аналіз інформаційних потоків та оцінено інтенсивність обробки даних. Побудовано логічну модель комплексу та комплекс діаграм. Сформульовано технічні та функціональні вимоги до програмного продукту.

Розроблений програмно-методичний комплекс дозволяє мінімізувати інженерні помилки на етапі проєктування, забезпечити технічно обґрунтований підбір обладнання без залучення вузькопрофільних фахівців, підвищити надійність, безпеку та економічну ефективність приватних сонячних електростанцій.

КЛЮЧОВІ СЛОВА: ПРОГРАМНО-МЕТОДИЧНИЙ КОМПЛЕКС, ІНФОРМАЦІЙНА ПІДТРИМКА, СОНЯЧНІ ПАНЕЛІ, ІНВЕРТОРИ, ЕЛЕКТРИЧНА СУМІСНІСТЬ, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, ПОБУТОВА ЕНЕРГЕТИКА, ПІДБІР ОБЛАДНАННЯ.

ABSTRACT

Tyshchenko S.A. Software-methodological complex for information support of the process of selecting solar panels and inverters for alternative power supply of households in Ukraine.

Qualification work for the degree of Bachelor of Computer Science, specialty 122 "Computer Science". – LLC "TECHNICAL UNIVERSITY "METINVEST POLITECHNIKA" of the Ministry of Education and Science of Ukraine, Zaporizhzhia, 2026.

The purpose of the work was to create a software-methodological complex that provides information support and automation of the process of selecting solar panels and inverters for the alternative power supply of private households in Ukraine. The object of the study is the process of designing and selecting components for domestic photovoltaic systems.

The subject of the study is the methods and software tools for information support of the technical matching of photovoltaic modules with power converters based on the modeling of electrical and energy parameters.

Methodology – systems analysis of the subject area, mathematical modeling of the energy balance, object-oriented design, methods for verifying electrical compatibility, algorithmization of configuration selection, UML/SADT modeling of information flows.

The work analyzes the state of the domestic solar energy market, defines a development scenario for the subject area, and outlines key concepts for modeling. A mathematical model for verifying equipment compatibility based on technical criteria has been developed. A quantitative analysis of information flows was conducted, and the intensity of data processing was evaluated. A logical model of the complex and a comprehensive set of diagrams were constructed. Technical and functional requirements for the software product were formulated.

The developed software-methodological complex allows minimizing engineering errors at the design stage, providing a technically justified selection of equipment without involving highly specialized professionals, and increasing the reliability, safety, and economic efficiency of private solar power plants.

KEYWORDS: SOFTWARE-METHODOLOGICAL COMPLEX, INFORMATION SUPPORT, SOLAR PANELS, INVERTERS, ELECTRICAL COMPATIBILITY, MATHEMATICAL MODELING, DOMESTIC ENERGY, EQUIPMENT SELECTION.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРИНЦИПІВ МОДЕЛЮВАННЯ СИСТЕМИ ПІДБОРУ КОМПОНЕНТІВ СОНЯЧНОЇ ЕНЕРГЕТИКИ.....	8
1.1 Аналіз предметної області та обґрунтування актуальності дослідження в сфері альтернативної енергетики	8
1.2 Сценарій розвитку предметної області та ключові поняття для моделювання.....	9
1.3 Склад, структура системи та параметри моделювання.....	10
1.4 Аналіз сучасних принципів моделювання та огляд програмних аналогів.....	11
1.5 Глосарій предметної області для автоматизації підбору сонячних панелей та інверторів	16
1.6 Висновок до розділу 1	17
2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ПІДБОРУ КОМПОНЕНТІВ СОНЯЧНОЇ ЕНЕРГЕТИКИ	19
2.1 Побудова логічної моделі системи проєкту програмно-методичного комплексу для підбору сонячних панелей та інверторів.....	19
2.2 Моделювання інформаційних потоків системи.....	27
2.3 Розробка технічного завдання на створення програмного засобу	31
3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО КОМПЛЕКСУ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ПРОЦЕСУ ПІДБОРУ СОНЯЧНИХ ПАНЕЛЕЙ ТА ІНВЕРТОРІВ ДЛЯ АЛЬТЕРНАТИВНОГО ЖИВЛЕННЯ ДОМОГОСПОДАРСТВ УКРАЇНИ.....	34
3.1 Вибір та обґрунтування засобів розробки та технологічного стеку	34
3.1.1 Середовище розробки Visual Studio Code та принципи організації робочого процесу	34
3.1.2 Система контролю версій GitHub та організація командної взаємодії	36
3.1.3 Мова програмування JavaScript та середовище виконання Node.js.....	37

3.1.4 Фреймворк Electron для кросплатформної розробки десктопних застосунків	38
3.1.5 Система управління базами даних SQLite та ORM-бібліотека Prisma.....	39
3.2 Проектування структури програмно-методичного комплексу	41
3.2.1 Багатозарова архітектура та принципи поділу відповідальності	42
3.2.2 Організація кодової бази та структура каталогів проекту.....	43
3.2.3 Механізми міжмодульної взаємодії та керування конфігурацією	45
3.2.4 Проектування реляційної моделі даних та ER-діаграма	46
3.2.5 Об'єктно-орієнтоване моделювання системи та UML-діаграма класів.....	49
3.3 Реалізація програмного комплексу та інтерфейсу користувача	52
3.3.1 Головне вікно калькулятора.....	53
3.3.2 Модальне вікно характеристик фотомодуля.....	58
3.3.3 Історія перевірок.....	60
3.3.5 Панель адміністратора.....	62
4 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ТА ВИКОРИСТАННЯ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ	70
4.1 Розрахунок витрат на розробку програмного комплексу	70
4.2 Оцінка конкурентоспроможності розробленого програмного забезпечення.....	80
4.3 Аналіз економічного ефекту від впровадження програмного продукту..	84
ВИСНОВКИ	88
ПЕРЕЛІК ПОСИЛАНЬ	90
ДОДАТОК А. ВІДОМОСТІ РОБОТИ	97
ДОДАТОК Б. ГЛОСАРІЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСУ ПІДБОРУ СОНЯЧНИХ ПАНЕЛЕЙ ТА ІНВЕРТОРІВ ДЛЯ АЛЬТЕРНАТИВНОГО ЖИВЛЕННЯ ДОМОГОСПОДАРСТВ УКРАЇНИ	98
ДОДАТОК В. ТЕХНІЧНЕ ЗАВДАННЯ	100
ДОДАТОК Г. АПРОБАЦІЯ РЕЗУЛЬТАТІВ	108

ВСТУП

Сучасний стан та тенденції розвитку енергетичного сектору України характеризуються високим рівнем турбулентності. Масштабні пошкодження об'єктів критичної інфраструктури, дефіцит генеруючих потужностей в об'єднаній енергосистемі та, як наслідок, регулярні обмеження споживачів зумовлюють гостру потребу у забезпеченні енергетичної незалежності на рівні окремих побутових споживачів [1]. У цих умовах розгортання сонячних електростанцій для приватних домогосподарств стає пріоритетним напрямом розвитку альтернативної енергетики.

Ефективність функціонування домашньої СЕС безпосередньо залежить від етапу її проектування та конфігурування. Процес індивідуального підбору обладнання – фотоелектричних модулів та гібридних інверторів є складною багатокритеріальною інженерною задачею. Необхідно одночасно враховувати індивідуальні профілі енергоспоживання об'єкта, номінальну потужність пристроїв, а також жорсткі технічні обмеження сумісності компонентів за струмом та напругою для забезпечення стабільної роботи системи[2].

На практиці більшість споживачів або дрібних компаній-інсталяторів здійснюють підбір обладнання на основі спрощених чи усереднених розрахунків. Це призводить або до надлишкових інвестицій, або до неефективної роботи системи через технічну неузгодженість компонентів. Наявні професійні програмні комплекси є дорогішими, складними для пересічного користувача та потребують тривалого освоєння. Безкоштовні онлайн-калькулятори мають закриті алгоритми, фіксований набір обладнання та маркетингову упередженість.

Метою даної роботи є розробка програмного забезпечення, яке автоматизує процес підбору сонячних панелей та гібридних інверторів для систем альтернативного живлення приватних домогосподарств, забезпечує перевірку електричної сумісності технічних компонентів станції та надає

інформаційну підтримку користувачу під час прийняття рішень щодо оптимальної конфігурації обладнання.

У межах проєкту передбачено:

- проведення комплексного аналізу предметної області, існуючих методик розрахунку сонячних електростанцій та аналогів програмного забезпечення;

- проєктування архітектури та структури реляційної бази даних для збереження інформації про технічні характеристики обладнання та користувацькі конфігурації;

- розробка алгоритмів перевірки технічної сумісності фотоелектричних модулів та гібридних інверторів за параметрами струму та напруги;

- розробка кросплатформного програмного застосунку мовою JavaScript з використанням середовища розробки Visual Studio Code;

- створення користувацького інтерфейсу для зручного введення вхідних даних, відображення результатів підбору та адміністрування бази даних пристроїв;

- проведення тестування розробленого програмного забезпечення для підтвердження його працездатності та точності розрахунків.

Таким чином, розробка програмно-методичного комплексу адаптованого до умов України, має стати ефективним інструментом у щоденній роботі інсталляторів сонячних систем та є актуальним науково-практичним завданням у галузі прикладних комп'ютерних наук.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРИНЦИПІВ МОДЕЛЮВАННЯ СИСТЕМИ ПІДБОРУ КОМПОНЕНТІВ СОНЯЧНОЇ ЕНЕРГЕТИКИ

1.1 Аналіз предметної області та обґрунтування актуальності дослідження в сфері альтернативної енергетики

Сучасний етап розвитку енергетичної інфраструктури України характеризується стрімким зростанням попиту на автономні та розподілені джерела електропостачання. Децентралізація генеруючих потужностей, нестабільність централізованої мережі та економічні фактори сприяють масовому переходу домогосподарств до використання відновлювальних джерел енергії [3]. Сонячна енергетика посідає провідну роль у цьому процесі завдяки доступності технологій, прогнозованості генерації та відносно низьким експлуатаційним витратам. Ключовим етапом створення ефективної сонячної електростанції для побутового споживання є правильний підбір основних компонентів, зокрема сонячних панелей та інверторів. Від точності цього підбору безпосередньо залежать продуктивність системи, термін її служби та економічна віддача.

На практиці процес проектування приватних сонячних станцій часто супроводжується використанням застарілих методик, ручних розрахунків або універсальних калькуляторів, які не враховують специфіку українського клімату, локальні тарифи на електроенергію та технічні особливості побутового споживання [4]. Це призводить до низки системних проблем. Неправильне узгодження потужності панелей та характеристик інвертора знижує загальний коефіцієнт корисної дії системи. Ігнорування сезонних змін інсоляції призводить до значного розриву між очікуваною та фактичною генерацією. Відсутність інтегрованого аналізу навантаження домогосподарства ускладнює прогнозування енергетичної автономності. Дані

фактори безпосередньо впливають на фінансові показники проекту та термін його окупності.

Автоматизація процесу підбору обладнання через спеціалізований програмно-методичний комплекс дозволяє усунути зазначені недоліки. Така система має забезпечувати точне математичне моделювання енергетичного балансу, автоматичний розрахунок втрат, оптимізацію співвідношення потужності генеруючих модулів та перетворювачів, а також формування обґрунтованих рекомендацій для кінцевого користувача [5], [6]. Вимоги до програмно-методичного комплексу як інструмента моделювання передбачають забезпечення точності розрахунків енергетичного балансу з допустимою похибкою не більше п'яти відсотків, підтримку локалізованої бази метеорологічних даних для території України, можливість зміни вхідних параметрів навантаження та географічного розташування в реальному часі, автоматичну перевірку сумісності номінальних характеристик сонячних модулів та інверторів, а також формування підсумкової технічної специфікації у зручному для користувача форматі. Актуальність дослідження полягає у створенні інструменту, який поєднує інженерну точність з доступністю інтерфейсу, що відповідає сучасним вимогам цифровізації побутової енергетики.

1.2 Сценарій розвитку предметної області та ключові поняття для моделювання

Розвиток ринку побутової сонячної енергетики в Україні пройшов кілька етапних трансформацій. На початку двохтисячних років технологія мала експериментальний характер і використовувалася переважно в ізольованих регіонах [10]. З впровадженням механізмів зеленого тарифу у 2009 році розпочався період активного ринкового зростання [11]. Власники

домогосподарств орієнтувалися на максимізацію встановленої потужності для продажу надлишків електроенергії в мережу. Цей етап характеризувався переважанням мережевих інверторів та стандартних кремнієвих модулів. Після зміни законодавчого регулювання та переходу на механізм чистого вимірювання пріоритети змістилися у бік оптимізації власного споживання [12]. Сучасний етап розвитку орієнтований на підвищення енергоефективності, точний підбір обладнання під індивідуальні навантаження та мінімізацію втрат при перетворенні постійного струму в змінний.

Для подальшого математичного та програмного моделювання необхідно чітко визначити базові поняття предметної області. Сонячна панель розглядається як перетворювач сонячного випромінювання в електричну енергію з фіксованими номінальними параметрами потужності, напруги відкритого кола та струму короткого замикання. Інвертор класифікується як силовий електронний пристрій, що забезпечує узгодження параметрів генерації з вимогами споживчої мережі або навантаження. Профіль споживання домогосподарства визначає розподіл електроенергії за годинами доби та місяцями року. Індекс інсоляції відображає кількість сонячної радіації, що надходить на одиницю площі в конкретному географічному регіоні. Коефіцієнт узгодження потужностей характеризує співвідношення сумарної потужності масиву панелей до номінальної потужності інвертора. Ці поняття формують логічну основу для побудови алгоритмічної моделі комплексу.

1.3 Склад, структура системи та параметри моделювання

Для побудови коректної інформаційної моделі необхідно виділити базовий понятійний апарат предметної області. До основних понять належать: номінальна потужність, напруга відкритого контуру, робочий струм, діапазон

відстеження точки максимальної потужності, сумарна потужність навантаження та електрична сумісність [7], [8].

Досліджувана система складається з двох взаємопов'язаних підсистем: генеруючої та перетворювальної.

Генеруюча підсистема сформована масивом фотоелектричних модулів, з'єднаних між собою у послідовні чи паралельні ланцюги.

Перетворювальна підсистема представлена гібридним інвертором, що містить у своєму складі контролер заряду та блок перетворення постійного струму в змінний.

Структурні зв'язки між компонентами мають жорсткий характер. Постійний струм, згенерований панелями, надходить на вхід контролера інвертора [9]. Інвертор, відповідно до поточного рівня споживання домогосподарства, трансформує необхідну частину енергії та передає її на внутрішню розподільну мережу будівлі. Математична цілісність цієї структури забезпечується виконанням системи нерівностей, які обмежують вихідні параметри масиву панелей рамками допустимих вхідних значень інвертора.

1.4 Аналіз сучасних принципів моделювання та огляд програмних аналогів

Сучасні принципи моделювання енергетичних систем базуються на поєднанні детермінованих розрахунків з ймовірнісними оцінками кліматичних факторів. Детерміновані моделі використовують фізичні закони перетворення енергії та паспортні дані обладнання для розрахунку очікуваної генерації. Ймовірнісні підходи застосовують статистичну обробку метеорологічних даних за тривалі періоди спостережень для оцінки ризиків відхилень від планових показників [13]. Методика багатокритеріальної оптимізації дозволяє знаходити баланс між технічною продуктивністю, капітальними витратами та терміном

окупності. Для інформаційної підтримки процесу підбору найбільш ефективним є гібридний підхід, що поєднує фізичне моделювання енергетичного балансу з алгоритмами перебору сумісних компонентів.

На ринку існує декілька категорій програмних рішень, що використовуються для проектування та підбору обладнання сонячних електростанцій. Для порівняння було обрано три популярні системи: PVsyst, HelioScope та SolarEdge Design Studio.

Детальний огляд PVsyst та виявлення його переваг та недоліків.

На рисунку 1.1 наведено діалогове вікно системи PVsyst [13].

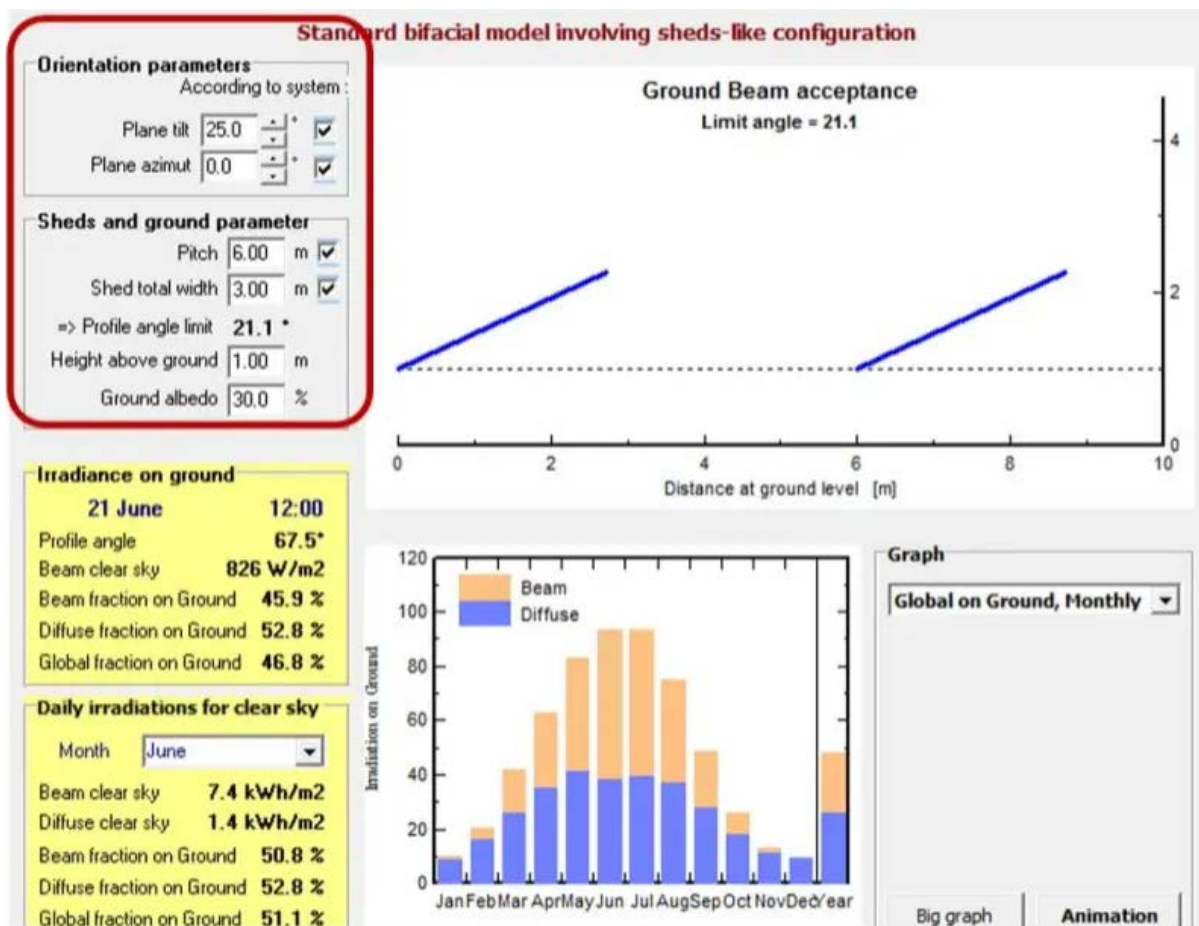


Рисунок 1.1 – Діалогове вікно системи PVsyst

PVsyst є професійним програмним комплексом швейцарської розробки, який вважається індустріальним стандартом для інженерів-проектувальників великих промислових сонячних електростанцій [14]. Програмний пакет

призначений для точного моделювання фотоелектричних систем та розрахунку річної виробітки електроенергії з урахуванням географічного розташування об'єкта, орієнтації масиву, температурних коефіцієнтів, втрат у кабельних мережах і впливу зовнішніх факторів. Математичне забезпечення комплексу базується на детальному моделюванні фізичних процесів у напівпровідникових переходах, аналізі вольт-амперних характеристик фотомодулів та розрахунку термодинамічних втрат. Система використовує глобальні бази метеорологічних даних, підтримує тривимірне моделювання затінення об'єктів, а також містить розширену бібліотеку компонентів із тисячами моделей сонячних панелей та інверторів різних виробників. Алгоритми узгодження потужностей дозволяють автоматично перевіряти сумісність параметрів фотоелектричного масиву з діапазоном точки максимальної потужності інвертора. Крім технічного моделювання, програма забезпечує розрахунок економічних показників та формування звітів відповідно до міжнародних стандартів [15]. Водночас інтерфейс платформи характеризується значною кількістю спеціалізованих фізичних параметрів і коефіцієнтів, що ускладнює її використання для швидкого індивідуального підбору обладнання пересічними користувачами побутового рівня. Переваги та недоліки системи PVsyst наведено у таблиці 1.1.

Таблиця 1.1 – Переваги та недоліки системи PVsyst

Переваги	Недоліки
Висока точність математичного моделювання	Надзвичайно висока вартість комерційної ліцензії
Розширена база даних обладнання	Складний інтерфейс для новачків
Підтримка міжнародних стандартів проектування	Орієнтація на промислові об'єкти

Детальний огляд HelioScore та виявлення його переваг та недоліків. На рисунку 1.2 наведено діалогове вікно системи HelioScore

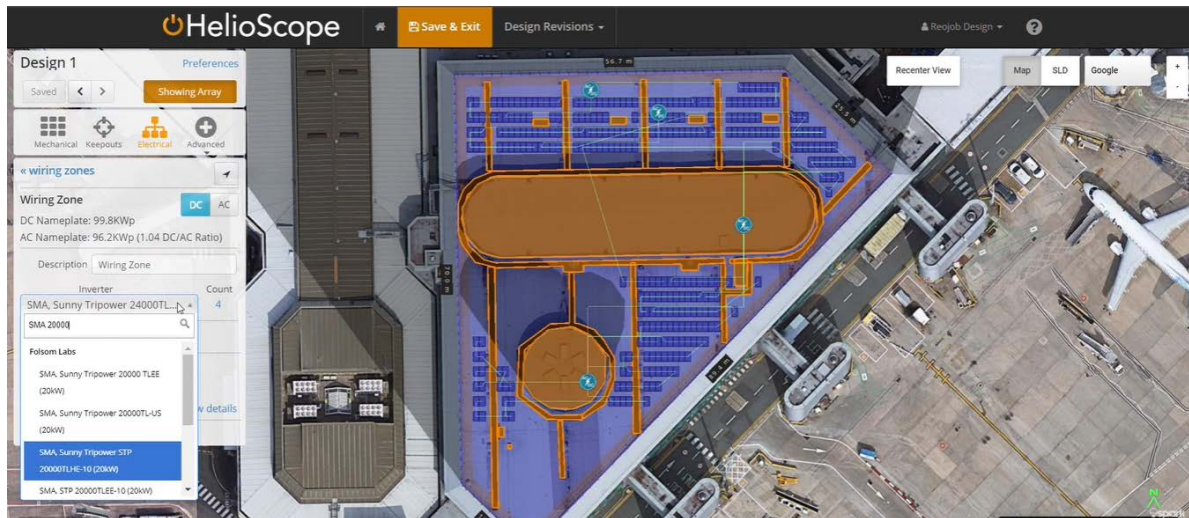


Рисунок 1.2 – Діалогове вікно системи HelioScore

HelioScore представляє собою хмарну платформу для швидкого проектування сонячних електростанцій. Система інтегрує інструменти для автоматичного розміщення панелей на плані даху з використанням супутникових знімків та цифрових моделей рельєфу [16]. Програма забезпечує миттєвий розрахунок очікуваної генерації та автоматичний підбір інверторів на основі заданих параметрів навантаження. Функціонал включає бібліотеку сумісних компонентів, алгоритми оптимізації розташування масиву та генератор комерційних пропозицій. Платформа орієнтована на монтажні організації та інтегратори, що забезпечує високу швидкість підготовки проєктної документації. Переваги та недоліки системи HelioScore наведено у таблиці 1.2.

Таблиця 1.2 – Переваги та недоліки системи HelioScore

Переваги	Недоліки
Інтуїтивний графічний інтерфейс	Потребує стабільного інтернет-з'єднання
Швидке автоматичне розташування панелей	Обмежена підтримка локальних тарифів
Вбудований генератор комерційних пропозицій	Підписка моделі обслуговування

Детальний огляд SolarEdge Design Studio та виявлення його переваг та недоліків. На рисунку 1.3 наведено діалогове вікно системи SolarEdge Design Studio



Рисунок 1.3 – Діалогове вікно системи SolarEdge Design Studio

SolarEdge Design Studio є спеціалізованим конфігуратором, розробленим виробником силових електронних рішень. Інструмент зосереджений на підборі оптимальної конфігурації інверторів та оптимізаторів потужності для конкретних сонячних модулів [17]. Система автоматично перевіряє електричну сумісність компонентів, розраховує втрати напруги та генерує схему підключення стрічок. Програма містить базу даних сумісних панелей, дозволяє моделювати вплив часткового затінення та формує детальну специфікацію обладнання. Інструмент інтегрується з фірмовим обладнанням, що забезпечує високу точність розрахунків для конкретної архітектури мережі.

Переваги та недоліки системи SolarEdge Design Studio наведено у таблиці 1.3.

Таблиця 1.3 – Переваги та недоліки системи SolarEdge Design Studio

Переваги	Недоліки
Точний розрахунок електричної сумісності	Прив'язка до обладнання одного бренду
Автоматична перевірка відповідності нормам	Обмежений вибір сторонніх компонентів
Детальна специфікація та схема підключення	Відсутність аналітики для побутового споживача

Висновок. Проведений аналіз існуючих програмних рішень дозволяє дійти висновку, що жодне з них не забезпечує повного задоволення потреб домогосподарств України. Професійні пакети відрізняються надмірною складністю та високою вартістю, хмарні платформи вимагають постійного підключення до мережі та не адаптовані до локальних умов, а фірмові конфігуратори обмежують вибір обладнання конкретним виробником [18].

Також виявлено відсутність інструментів, орієнтованих на простий підбір панелей та інверторів без залучення акумуляторних систем, з урахуванням українських кліматичних даних та побутових навантажень. Ураховуючи вищезазначене, можна стверджувати про доцільність розробки власного програмно-методичного комплексу, який максимально відповідатиме вимогам підбору обладнання для альтернативного живлення домогосподарств, забезпечуватиме технічну точність та зручність використання без надлишкових функцій.

1.5 Глосарій предметної області для автоматизації підбору сонячних панелей та інверторів

Глосарій предметної області для автоматизації підбору сонячних панелей та інверторів розміщено у додатку Б.

1.6 Висновок до розділу 1

У першому розділі кваліфікаційної роботи виконано комплексний науково-технічний аналіз стану питання та базових концепцій, пов'язаних із проблемою інформаційної підтримки та автоматизації процесу індивідуального підбору фотоелектричного обладнання. Проведене дослідження дозволило обґрунтувати високу актуальність розробки у зв'язку з децентралізацією об'єднаної енергетичної системи України та масовим переходом приватних домогосподарств на альтернативні джерела живлення. Було встановлено, що хаотичний та інтуїтивний підбір технічних засобів без належної інженерної перевірки сумісності призводить до критичних перевантажень обладнання та аварійних ситуацій, що робить створення спеціалізованих локальних IT-інструментів першочерговою практичною потребою.

Під час аналізу аспектів розгляду та моделювання заданої системи сонячну станцію було формалізовано як структуру взаємопов'язаних підсистем, а саме генеруючого масиву сонячних панелей та перетворювального гібридного інвертора. Визначено, що математичне та алгоритмічне моделювання цієї системи має базуватися на суворому контролі обмежень електричних параметрів, серед яких визначальними є діапазони напруги відкритого контуру, струму короткого замикання та робочих зон контролера відстеження точки максимальної потужності. Огляд існуючих аналогів, включаючи важкі інженерні комплекси проектування та спрощені маркетингові онлайн-віджети, підтвердив наявність суттєвого технологічного розриву, оскільки на ринку відсутні доступні автономні інструменти для точного інженерного аналізу сумісності пристроїв без прив'язки до хмарних закордонних сервісів.

На основі виявлених недоліків існуючого програмного забезпечення було сформульовано чіткі вимоги до програмно-методичного комплексу як

інструмента моделювання. Система має забезпечувати повноцінний інтерфейс для виконання операцій керування даними технічних пристроїв у реляційній базі даних, автоматично розраховувати граничні межі кількості фотомодулів у ланцюгах та валідувати конфігурації за критеріями сумісності струму й напруги. З технічного погляду обґрунтовано доцільність реалізації комплексу як локального кросплатформного додатка мовою Javascript у середовищі розробки Visual Studio Code, що гарантує стабільність, високу швидкість, цілісність збереження інформації та повну незалежність від зовнішніх мережевих ресурсів. Отримані теоретичні висновки є безпосереднім підґрунтям для подальшого проектування архітектури та алгоритмічного забезпечення системи у наступних розділах роботи.

2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ПІДБОРУ КОМПОНЕНТІВ СОНЯЧНОЇ ЕНЕРГЕТИКИ

2.1 Побудова логічної моделі системи проекту програмно-методичного комплексу для підбору сонячних панелей та інверторів

Для візуалізації архітектури та функціонування програмно-методичного комплексу було застосовано комплекс діаграмних методик, що охоплюють як структурні, так і поведінкові аспекти системи. Кожна діаграма виконує визначену роль у процесі проектування та дозволяє розкрити конкретний аспект функціонування системи.

Для представлення функціональних модулів було застосовано SADT-діаграми [19]. Відповідно до методології SADT, об'єкт аналізу поділяється на окремі логічні підпроцеси, для яких визначаються вхідні та вихідні дані, механізми реалізації, а також елементи керування.

SADT (від англ. Structured Analysis and Design Technique) є методологією аналізу та проектування систем [20]. Вона широко використовувалася у військовій, промисловій і комерційній сферах для розв'язання різноманітних завдань, зокрема:

- розроблення програмного забезпечення телефонних мереж;
- створення систем підтримки та діагностики;
- стратегічного й довгострокового планування;
- автоматизації виробничих процесів і проектування;
- конфігурації комп'ютерних комплексів;
- підготовки та навчання персоналу;
- створення вбудованого програмного забезпечення для оборонних систем;

- управління фінансовими ресурсами та матеріально-технічним забезпеченням.

- Серед сучасних методів опису систем SADT вирізняється широкими можливостями застосування, оскільки:

- забезпечує наочне відображення таких характеристик системи, як керування, виконавці та зворотний зв'язок;

- підтримує організацію колективної роботи під час створення систем;

- орієнтована на використання на початкових етапах проєктування;

- може інтегруватися з іншими структурними методами завдяки використанню графічних моделей SADT як засобу поєднання різних підходів до опису системи на різних рівнях деталізації [21].

Основними правилами побудови SADT-діаграм є:

- обмеження кількості функціональних блоків на одному рівні декомпозиції (зазвичай від 3 до 6 блоків);

- забезпечення зв'язності між діаграмами за допомогою нумерації блоків;

- використання унікальних назв і позначень без повторень;

- дотримання встановлених синтаксичних правил побудови блоків і дуг;

- чітке розмежування вхідних даних та елементів керування;

- відокремлення організаційної структури від функціональної моделі системи.

На рисунку 2.1 надана SADT-діаграма 0 - рівня процесу А-0 в якій система підбору обладнання компонентів сонячної енергетики представлена як єдиний функціональний блок, що взаємодіє із зовнішнім середовищем. Опис діаграми наведено у таблиці 2.1.

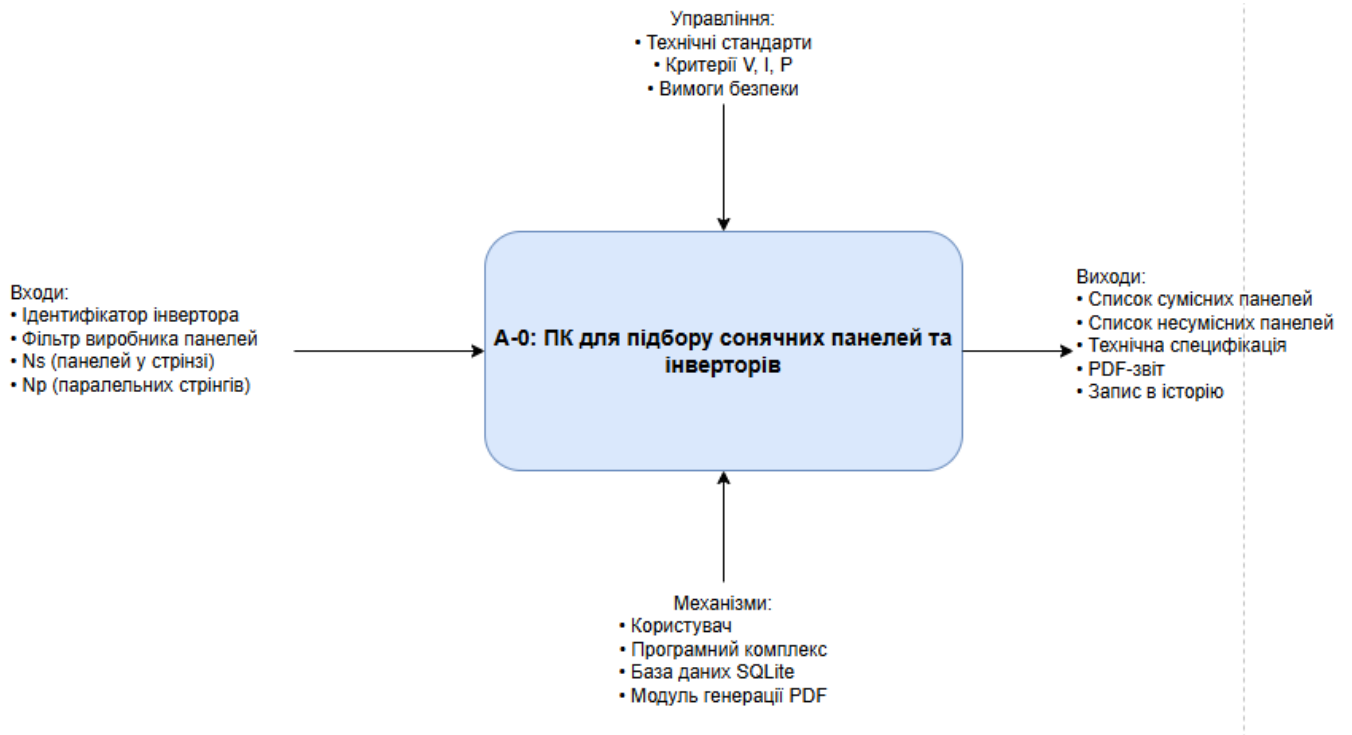


Рисунок 2.1 – SADT-діаграма 0-рівня процесу A-0 «ПК системи підбору сонячних панелей та інверторів»

Таблиця 2.1 – Опис SADT-діаграми 0-рівня процесу A-0

Елемент	Зміст
Найменування процесу	ПК для підбору сонячних панелей та інверторів
Вхідні дані	Ідентифікатор обраного інвертора; фільтр за виробником панелей; кількість панелей у стрінзі (Ns); кількість паралельних стрінгів (Nr)
Управління	Технічні стандарти сумісності; правила перевірки напруги, струму та потужності; вимоги до безпеки експлуатації
Механізми	Користувач (продавець-інсталятор або проектувальник); програмний комплекс; база даних обладнання; модуль генерації PDF
Вихідні дані	Список сумісних панелей; список несумісних панелей із причинами відхилення; технічна специфікація підбраного обладнання; PDF-звіт; запис в історію розрахунків

На діаграмі першого рівня процес А-0 декомпоновано на три основні функціональні блоки, кожен з яких виконує визначену підзадачу в межах загального процесу підбору обладнання. На рисунку 2.2 представлена деталізуюча SADT-діаграма першого рівня.

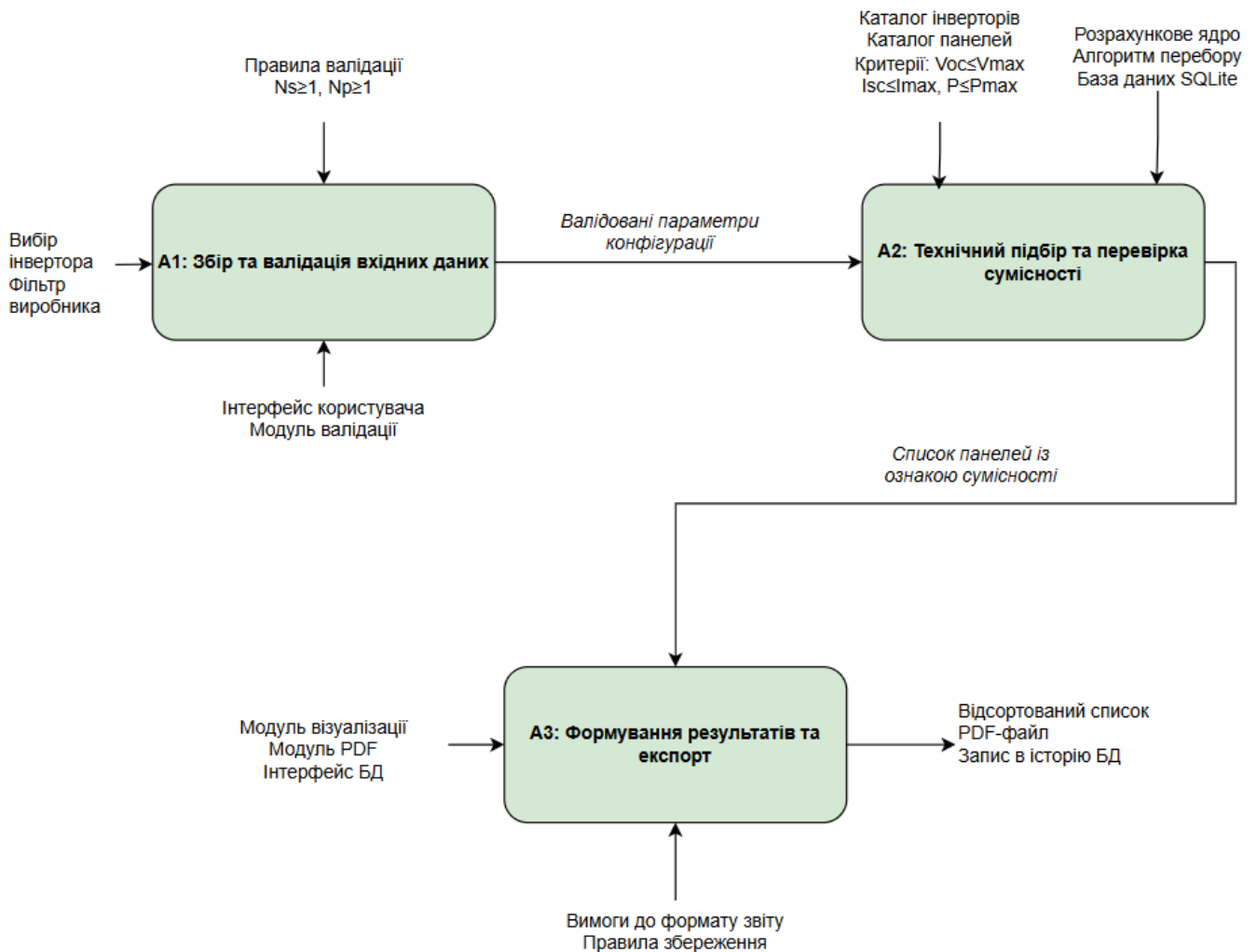


Рисунок 2.2 – Деталізуюча SADT-діаграма першого рівня

Блок А1: «Збір та валідація вхідних даних». Цей блок відповідає за прийом параметрів конфігурації від користувача та перевірку їх коректності. На вхід блоку надходять: вибір інвертора з каталогу, фільтр за виробником панелей, значення N_s та N_r . Керуючими факторами є правила валідації ($N_s \geq 1$, $N_r \geq 1$, цілочисельні значення). Механізмами виконання є графічний

інтерфейс користувача та модуль перевірки даних. Вихідними даними блоку є валідовані параметри конфігурації, готові до передачі в модуль розрахунку.

Блок А2: «Технічний підбір та перевірка сумісності». Це центральний обчислювальний блок системи, що виконує основну функцію підбору. На вхід блоку надходять валідовані параметри конфігурації та довідкові дані з бази (характеристики інверторів та панелей). Керуючими факторами є математичні критерії сумісності: умова за напругою ($V_{oc_system} \leq V_{max}$), умова за струмом ($I_{sc_system} \leq I_{max}$), умова за потужністю ($P_{system} \leq P_{max}$). Механізмами є розрахункове ядро програми та алгоритм перебору сумісних конфігурацій. Вихідними даними є структурований список панелей із ознакою сумісності та пояснювальними повідомленнями.

Блок А3: «Формування результатів та експорт». Фінальний блок відповідає за візуалізацію результатів та їх збереження. На вхід блоку надходять результати перевірки сумісності. Керуючими факторами є вимоги до формату звіту та правила збереження даних. Механізмами є модуль відображення даних, модуль генерації PDF та інтерфейс взаємодії з базою даних. Вихідними даними є відсортований список панелей у графічному інтерфейсі, технічна специфікація, збережений запис в історії розрахунків та експортований PDF-файл.

Зв'язки між блоками. Блок А1 передає валідовані параметри конфігурації до блоку А2 для виконання розрахунків. Блок А2, у свою чергу, передає результати перевірки сумісності до блоку А3 для візуалізації та експорту. Усі блоки взаємодіють із єдиною базою даних, що забезпечує цілісність довідкових даних та історії розрахунків.

Опис діаграми наведено в таблиці 2.2

Таблиця 2.2 – опис SADT-діаграми першого рівня

№ блок у	Найменування операції	Вхідні дані	Управління	Виконавець	Вихідні дані
A1	Збір та валідація вхідних даних	Вибір інвертора, фільтр виробника, N_s, N_p	Правила валідації ($N_s \geq 1, N_p \geq 1$)	Користувач, інтерфейс програми	Валідовані параметри конфігурації
A2	Технічний підбір та перевірка сумісності	Валідовані параметри, каталог обладнання	Критерії сумісності (V, I, P)	Розрахункове ядро програми	Список панелей із ознакою сумісності
A3	Формування результатів та експорт	Результати перевірки	Вимоги до формату звіту	Модуль візуалізації, модуль PDF	Графічний звіт, PDF-файл, запис в БД

Діаграма варіантів використання була побудована для визначення функціональних вимог до системи з точки зору кінцевого користувача [22]. Ця діаграма відображає взаємодію між акторами та функціями, які система надає. На рисунку 2.3 наведено діаграму варіантів використання.

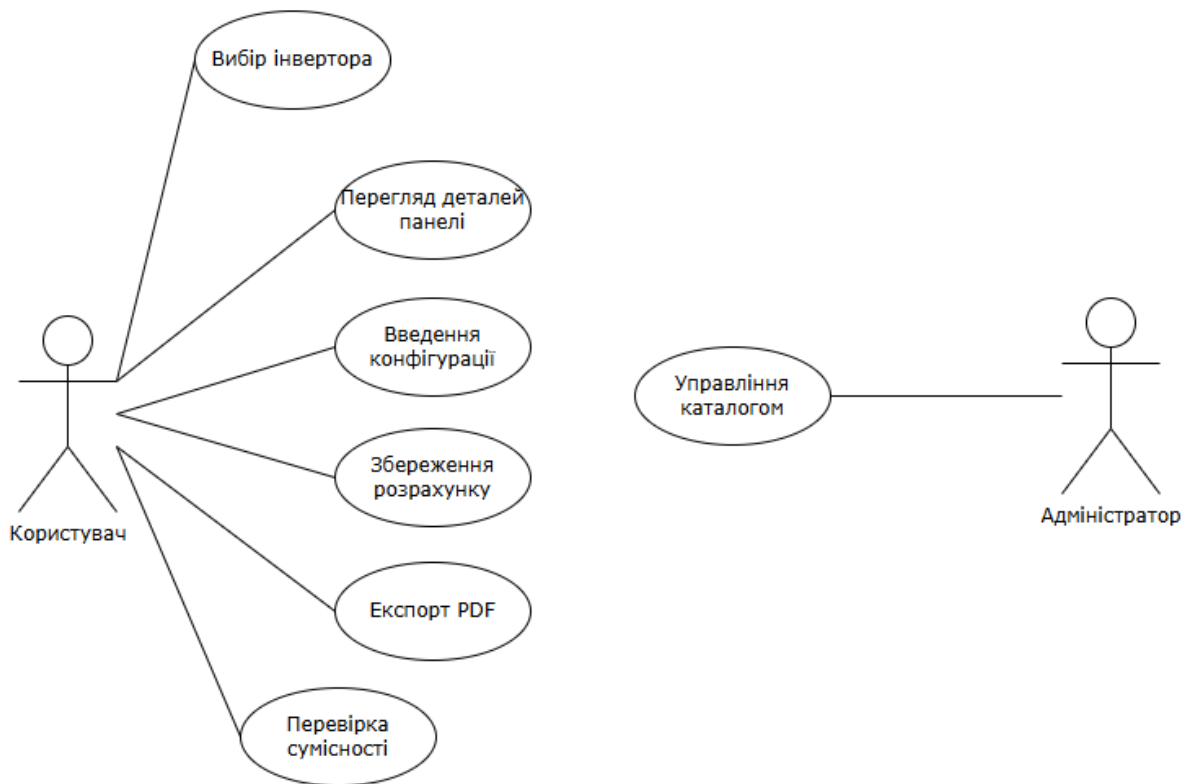


Рисунок 2.3 – Діаграма варіантів використання

У розроблюваній системі було виділено двох основних акторів: звичайного користувача який може бути інсталятором, проектувальником та адміністратора системи [23]. Для звичайного користувача визначено наступні варіанти використання: вибір інвертора з каталогу, введення конфігурації масиву панелей, перевірка сумісності обладнання, перегляд детальних характеристик панелі, збереження розрахунку в історію, експорт результатів у форматі PDF. Адміністратор системи має додаткові можливості: управління каталогом інверторів, зокрема додавання, редагування, видалення; управління каталогом сонячних панелей; перегляд історії розрахунків усіх користувачів. Діаграма варіантів використання дозволила чітко визначити межі системи, ідентифікувати основні функції та розмежувати права доступу для різних категорій користувачів.

Також, під час проектування програмного комплексу, було розроблено діаграму активності яка наведена на рисунку 2.4.

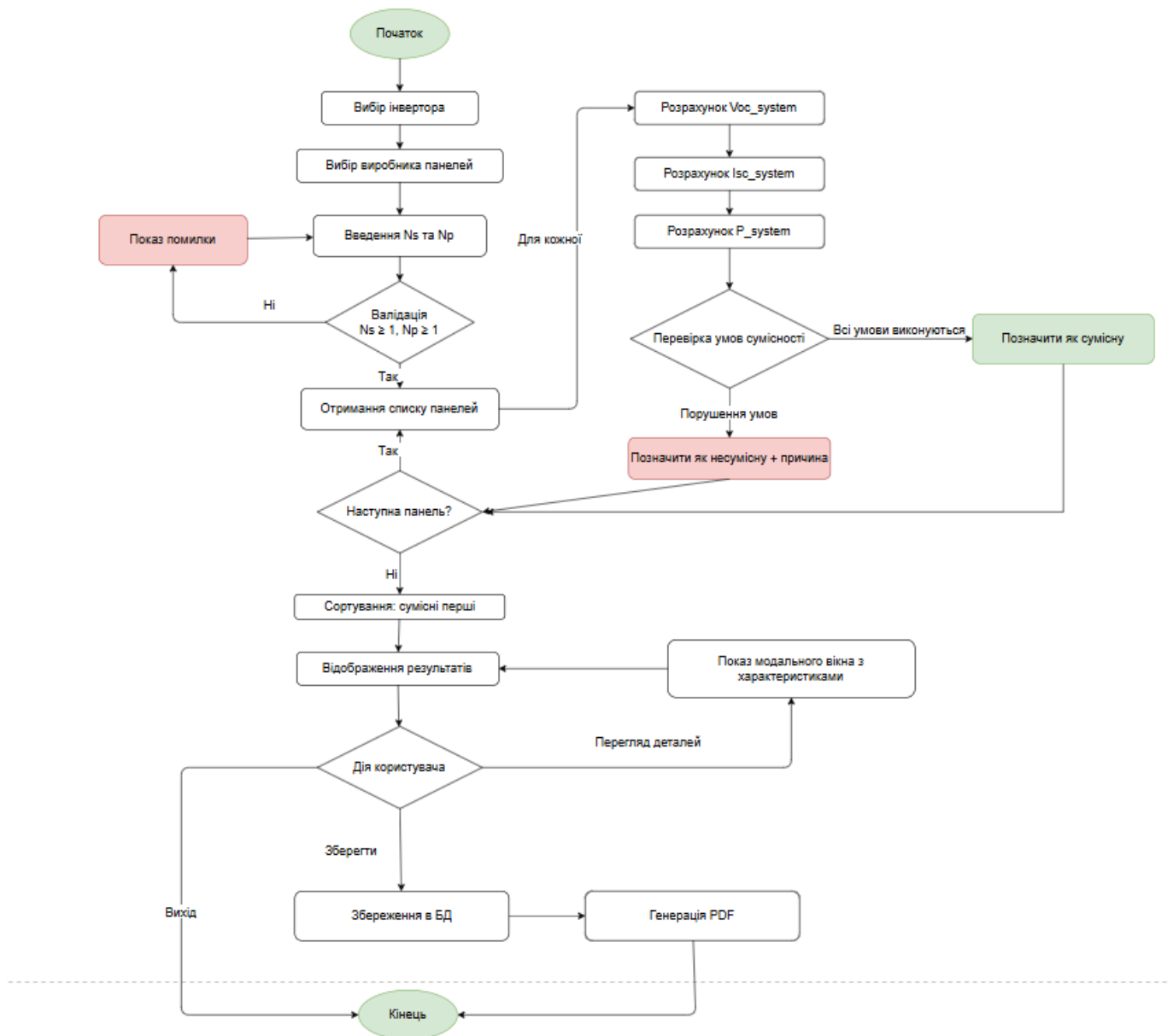


Рисунок 2.4 – Діаграма активності

Діаграма активності відображає потік управління від початкової дії до кінцевої, включаючи умовні переходи, паралельні процеси та цикли [24]. У розроблюваній системі діаграма активності починається з ініціалізації програми та завантаження довідкових даних з бази. Користувач обирає інвертор зі списку доступних моделей, після чого вводить параметри конфігурації: кількість панелей у стрінзі (N_s) та кількість паралельних стрінгів (N_p). Система виконує валідацію введених даних (перевірка на цілі числа, більші або рівні одиниці). У разі некоректних даних виводиться повідомлення

про помилку, і користувач повертається до введення параметрів. Після успішної валідації система формує SQL-запит для отримання списку сонячних панелей з бази даних з урахуванням фільтру за виробником. Для кожної панелі зі списку виконується цикл розрахунків: обчислення системної напруги холостого ходу, обчислення системного струму короткого замикання, обчислення загальної потужності. Далі виконується перевірка трьох умов сумісності: $V_{oc_system} \leq V_{max}$ інвертора, $I_{sc_system} \leq I_{max}$ інвертора, $P_{system} \leq P_{max}$ інвертора. Якщо всі умови виконуються, панель позначається як сумісна, інакше – як несумісна з вказанням конкретної причини невідповідності. Після обробки всіх панелей результати сортуються: сумісні панелі виводяться першими, відсортовані за спаданням потужності, потім – несумісні. Відсортований список відображається користувачеві. Користувач може виконати одну з наступних дій: переглянути деталі панелі, при цьому відкривається модальне вікно з повною специфікацією; зберегти розрахунок записавши в базу даних, додатково генерується PDF-файл; виконати новий розрахунок; завершити роботу програми. Діаграма активності дозволила чітко визначити логіку прийняття рішень у системі та послідовність виконання операцій.

2.2 Моделювання інформаційних потоків системи

Ефективне проектування програмно-методичного комплексу неможливе без глибокого розуміння структури та характеристик інформаційних потоків, що циркулюють у системі. Інформаційні потоки відображають передачу даних між різними компонентами системи: модулями програми, користувачем, базою даних та зовнішніми джерелами інформації. Дослідження цих потоків дозволяє виявити вузькі місця в архітектурі системи, оптимізувати структуру бази даних та забезпечити необхідну продуктивність програмного комплексу.

Інформаційний потік визначається як сукупність повідомлень (документів, запитів, відповідей), що передаються в певному напрямку між учасниками інформаційної системи. Для кількісної характеристики інформаційних потоків використовуються такі параметри:

- інтенсивність потоку – обсяг інформації, що проходить через систему за одиницю часу (вимірюється в кілобайтах на день або бітах на секунду);
- частота виникнення – середня кількість повідомлень за визначений період;
- тип потоку – внутрішній (між модулями системи) або зовнішній (взаємодія з користувачем чи зовнішніми системами), вхідний або вихідний;
- призначення – довідкова інформація, командні дані, реєстраційна інформація, результати розрахунків.

Для кількісного аналізу інформаційних потоків у розроблюваній системі підбору сонячних панелей та інверторів використовується формалізована модель оцінки інтенсивності потоків, що базується на наступній формулі:

$$I = N \times V \times F \quad (2.1)$$

де I – інтенсивність інформаційного потоку (КБ/день);

N – середня кількість запитів за день;

V – середній обсяг одного повідомлення (КБ);

F – середня частота обробки одного повідомлення за день.

Розглянемо основні інформаційні потоки в системі підбору обладнання для сонячних електростанцій.

Перший потік – вхідні дані від користувача. Користувач вводить наступну інформацію: обраний інвертор (ідентифікатор з бази даних), фільтр за виробником сонячних панелей (опціонально), кількість панелей у стрінзі (N_s), кількість паралельних стрінгів (N_p). Припустимо, що система використовується

приблизно 20 разів на день ($N = 20$). Середній обсяг одного запиту становить близько 0.5 КБ ($V = 0.5$), оскільки передаються лише ідентифікатори та числові параметри. Частота обробки – 1 раз на день ($F = 1$). Тоді інтенсивність цього потоку становить:

$$I_{\text{вхід}} = 20 \times 0.5 \times 1 = 10 \text{ КБ/день.}$$

Другий потік – довідкові дані з бази даних. Система звертається до бази даних для отримання каталогу інверторів та сонячних панелей. Припустимо, що в базі даних зберігається 50 моделей інверторів та 200 моделей сонячних панелей. Середній розмір запису про інвертор становить 0.3 КБ, про панель – 0.4 КБ. Щоденно система виконує приблизно 20 запитів до бази даних. Тоді обсяг довідкових даних становить:

$$V_{\text{дов}} = 50 \times 0.3 + 200 \times 0.4 = 15 + 80 = 95 \text{ КБ.}$$

Інтенсивність потоку довідкових даних:

$$I_{\text{дов}} = 20 \times 95 \times 1 = 1900 \text{ КБ/день.}$$

Третій потік – результати розрахунків. Після виконання перевірки сумісності система формує список результатів, що включає для кожної панелі: ідентифікатор, назву, потужність, ознаку сумісності, причину несумісності (за необхідності). Припустимо, що середній розмір результату по одній панелі становить 1-2 КБ. При 20 запитах на день та середній кількості панелей у результаті 50 штук отримуємо:

$$I_{\text{рез}} = 20 \times 50 \times 1.5 = 1500 \text{ КБ/день.}$$

Четвертий потік – збереження історії розрахунків. Кожен збережений розрахунок містить: ідентифікатор користувача, ідентифікатори обраного інвертора та панелі, параметри конфігурації (N_s , N_p , загальна кількість панелей), шлях до PDF-файлу, дату створення. Середній розмір одного запису історії становить приблизно 0.5 КБ. При 20 збереженнях на день:

$$I_{\text{іст}} = 20 \times 0.5 \times 1 = 10 \text{ КБ/день.}$$

Загальна інтенсивність інформаційного обміну в системі становить:

$$\begin{aligned} I_{\text{заг}} &= I_{\text{вхід}} + I_{\text{дов}} + I_{\text{рез}} + I_{\text{іст}} = 10 + 1900 + 1500 + 10 = \\ &= 3420 \text{ КБ/день} \approx 3.3 \text{ МБ/день.} \end{aligned}$$

Цей показник має важливе практичне значення для визначення вимог до пропускної здатності каналів передачі даних. Хоча система працює локально, це важливо для можливого мережевого розгортання. Також показник важливий для:

- розрахунку необхідного обсягу пам'яті для кешування даних;
- планування частоти резервного копіювання бази даних, адже при щоденному обсязі 3.3 МБ річний обсяг складе понад 1.2 ГБ;
- оптимізації структури бази даних (індексація, нормалізація таблиць);
- визначення вимог до продуктивності системи обробки запитів.

Крім кількісних характеристик, важливо класифікувати інформаційні потоки за типами:

Внутрішні потоки – передача даних між модулями системи, наприклад, від модуля збору вхідних даних до модуля технічного підбору, від модуля розрахунків до модуля формування результатів.

Зовнішні вхідні потоки – дані, що надходять від користувача, а саме параметри конфігурації масиву панелей, вибір інвертора, фільтри.

Зовнішні вихідні потоки – результати роботи системи, зокрема списки сумісних та несумісних панелей, технічні специфікації, PDF-звіти.

Потоки зберігання – запис та читання даних з бази даних як то каталоги обладнання, історія розрахунків.

Моделювання інформаційних потоків дозволило визначити оптимальну архітектуру системи, де дані мінімально дублюються, а передача інформації між модулями здійснюється ефективним чином. Зокрема, було прийнято рішення про централізоване зберігання довідкових даних у базі даних з кешуванням часто використовуваних запитів, що зменшує навантаження на систему прискорює час відгуку.

2.3 Розробка технічного завдання на створення програмного засобу

На основі результатів аналізу предметної області та виконаного функціонального моделювання сформовано технічні вимоги до програмного комплексу автоматизованого підбору фотоелектричних панелей та перевірки їх сумісності з інверторним обладнанням [54], [64].

Функціональні вимоги передбачають реалізацію механізму вибору інвертора з каталогу обладнання із можливістю фільтрації за виробником, номінальною потужністю, діапазоном робочої напруги та іншими технічними характеристиками. Система повинна забезпечувати введення конфігурації фотоелектричного масиву із зазначенням кількості послідовно з'єднаних модулів та кількості паралельних стрінгів із подальшою перевіркою коректності введених значень відповідно до умови. Програмний комплекс повинен автоматично виконувати перевірку сумісності фотоелектричних панелей з обраним інвертором за критеріями відповідності напруги холостого ходу, струму короткого замикання та сумарної потужності фотоелектричного масиву допустимим технічним параметрам інвертора. Результати розрахунків

повинні відображатися у структурованому вигляді з поділом конфігурацій на сумісні та несумісні, а для випадків невідповідності система має надавати пояснення причин відхилення технічних параметрів. Також необхідно реалізувати можливість перегляду детальних характеристик фотоелектричних панелей у окремому інформаційному вікні інтерфейсу користувача. Система повинна підтримувати збереження історії виконаних розрахунків у базі даних із можливістю формування та експорту звітної документації у форматі PDF [34], [35]. Для адміністрування програмного комплексу має бути передбачений окремий модуль управління каталогами обладнання, який забезпечуватиме додавання, редагування та видалення записів.

Вимоги до надійності передбачають реалізацію механізмів валідації всіх вхідних даних з метою запобігання помилкам під час виконання розрахунків та обробки інформації. Програмний комплекс повинен підтримувати автоматичне збереження даних після виконання основних операцій користувача. Для забезпечення інформаційної безпеки необхідно реалізувати систему автентифікації та авторизації користувачів, що дозволить обмежити несанкціонований доступ до функціональних можливостей системи. Крім того, програмний комплекс повинен забезпечувати ведення журналу подій та дій користувачів для можливості подальшого аудиту та контролю роботи системи.

Вимоги до продуктивності визначають, що час реакції системи на запити користувача не повинен перевищувати двох секунд за стандартних умов експлуатації. Програмний комплекс має забезпечувати стабільну одночасну роботу не менше десяти користувачів без суттєвого зниження швидкодії. База даних повинна підтримувати зберігання щонайменше 10 000 записів історії розрахунків без деградації продуктивності системи.

Вимоги до технічного забезпечення передбачають використання операційних систем Windows 10/11, macOS або Linux [25]. Мінімальні апаратні характеристики повинні включати процесор рівня Intel Core i3 або аналогічний за продуктивністю, не менше 4 ГБ оперативної пам'яті та щонайменше 500 МБ вільного дискового простору. Для функціонування програмного комплексу

також необхідна наявність середовища виконання Node.js версії 18.0 або вище [26].

Вимоги до програмного забезпечення визначають використання мов програмування JavaScript або TypeScript [27]. Для створення кросплатформного десктопного застосунку доцільно застосовувати фреймворк Electron, а для організації взаємодії з базою даних – ORM Prisma [28], [30]. Як систему керування базами даних передбачено використання SQLite для локального зберігання інформації [32]. Формування та експорт PDF-документів повинні реалізовуватися за допомогою вбудованих засобів середовища Electron [31].

Вимоги до користувацького інтерфейсу передбачають створення інтуїтивно зрозумілого графічного середовища українською мовою [29], [40]. Інтерфейс програмного комплексу повинен підтримувати адаптивне відображення для екранів із роздільною здатністю не менше 1280×720 пікселів [36]. Для підвищення зручності користування необхідно реалізувати візуальне розділення сумісних та несумісних конфігурацій за допомогою кольорової індикації [33]. Також інтерфейс повинен забезпечувати швидкий пошук, сортування та фільтрацію обладнання за технічними характеристиками, а також надавати доступ до довідкової інформації щодо основних параметрів фотоелектричних модулів та інверторів [37], [38], [39].

3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО КОМПЛЕКСУ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ПРОЦЕСУ ПІДБОРУ СОНЯЧНИХ ПАНЕЛЕЙ ТА ІНВЕРТОРІВ ДЛЯ АЛЬТЕРНАТИВНОГО ЖИВЛЕННЯ ДОМОГОСПОДАРСТВ УКРАЇНИ

3.1 Вибір та обґрунтування засобів розробки та технологічного стеку

Ефективна реалізація програмно-методичного комплексу для підбору сонячних панелей та інверторів вимагає ретельного підбору інструментарію розробки, який забезпечить стабільність роботи, кросплатформність, зручність супроводу та відповідність сучасним стандартам інженерії програмного забезпечення [41]. Вибір технологічного стеку та середовища розробки базувався на аналізі функціональних вимог системи, обмежень апаратного забезпечення цільових пристроїв, необхідності автономної роботи без постійного підключення до мережі Інтернет, а також вимог до безпеки локальних даних. Для створення застосунку було обране середовище розробки Visual Studio Code, принципи організації командної роботи через систему контролю версій GitHub, а також проведено поглиблений аналіз кожного компонента обраного технологічного стеку з обґрунтуванням доцільності їх застосування.

3.1.1 Середовище розробки Visual Studio Code та принципи організації робочого процесу

Середовище розробки Visual Studio Code обрано як основний інструмент для написання, налагодження та структурованого керування кодом проєкту

[45]. Це рішення зумовлено сукупністю технічних переваг, які безпосередньо впливають на продуктивність розробки та якість кінцевого програмного продукту. Visual Studio Code є кросплатформним редактором коду з відкритим вихідним кодом, що підтримує миттєве індексування файлів, інтелектуальне автодоповнення синтаксису, вбудоване керування терміналом та інтегровані засоби відлагодження. Архітектура середовища базується на модульній системі розширень, що дозволяє адаптувати робочий простір під специфіку конкретного стеку технологій без перевантаження інтерфейсу зайвими компонентами.

Організація робочого процесу в середовищі Visual Studio Code реалізована через стандартизовану структуру каталогів проекту, яка розділяє логіку представлення, бізнес-правила, рівень доступу до даних та конфігураційні файли. Такий підхід забезпечує ізолюваність модулів, спрощує пошук помилкових конструкцій та дозволяє паралельно розробляти різні функціональні блоки без конфліктів залежностей. Вбудований менеджер задач дозволяє автоматизувати рутинні операції, такі як запуск сервера розробки, виконання міграцій бази даних, перевірка синтаксису та генерація документації. Інтегрований дебагер підтримує покрокове виконання коду, встановлення точок зупинки, перегляд стеку викликів та моніторинг стану змінних у реальному часі, що значно прискорює виявлення та усунення логічних помилок на етапі проектування.

Важливим аспектом використання середовища є підтримка стандартизованих конфігураційних файлів, які фіксують налаштування форматування коду, правила лінтингу та версії залежностей. Це забезпечує однорідність стилю написання програми незалежно від кількості учасників команди та гарантує сумісність проекту при розгортанні на різних робочих станціях. Налаштування робочого простору синхронізується через хмарні профілі розробника, що дозволяє миттєво відновити середовище після перевстановлення операційної системи або переходу на новий комп'ютер без втрати персоналізованих параметрів.

3.1.2 Система контролю версій GitHub та організація командної взаємодії

Для керування історією змін коду, координації роботи між учасниками команди та забезпечення цілісності проекту використано платформу GitHub [46]. Система контролю версій Git інтегрована в процес розробки як фундаментальний інструмент відстеження модифікацій, що дозволяє фіксувати кожну ітерацію розвитку програмного комплексу, повертатися до стабільних станів у разі виявлення критичних помилок та паралельно працювати над різними функціональними напрямками без блокування основної гілки розробки. На рисунку 3.1 представлено вікно робочого простору проекту в процесі побудови.

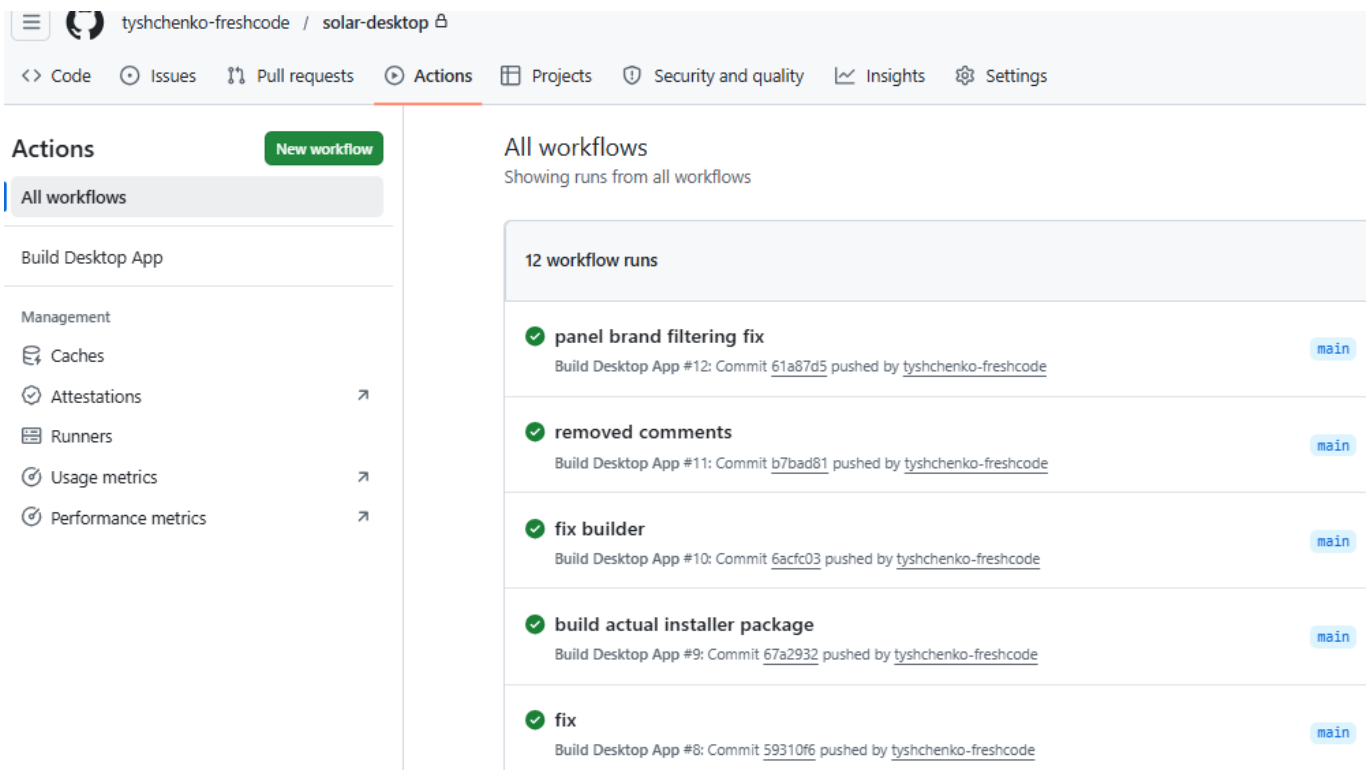


Рисунок 3.1 – Вікно робочого простору GitHub проекту

Організація репозиторію побудована на основі гілкової стратегії, яка передбачає виділення основної гілки для стабільних релізів, гілки розробки

для інтеграції нових функцій та тимчасових гілок для окремих завдань або виправлення помилок. Кожен фрагмент коду перед об'єднанням з основною кодовою базою проходить процедуру перевірки через запити на злиття, де реалізовується рецензування архітектурних рішень, аналіз безпеки викликів та перевірка відповідності стандартам кодування. Такий підхід мінімізує ризик внесення деструктивних змін у продуктивне середовище та забезпечує прозорість процесу прийняття технічних рішень.

Платформа GitHub додатково використовується для автоматизації процесів збірки та пакування застосунку. Конвеєри безперервної інтеграції налаштовані на автоматичний запуск тестування при кожному оновленні коду, перевірку сумісності залежностей та генерацію інсталяційних пакетів для операційних систем Windows та macOS. Конфігурація збірки включає підписування виконуваних файлів цифровими сертифікатами, що забезпечує довіру операційних систем до програмного продукту та попереджає блокування запуску вбудованими механізмами захисту. Система відстеження завдань інтегрована з репозиторієм для фіксації вимог, планування етапів розробки та документування виявлених дефектів, що створює єдиний інформаційний простір для керування життєвим циклом проєкту.

3.1.3 Мова програмування JavaScript та середовище виконання Node.js

Мова програмування JavaScript виступає основним засобом реалізації бізнес-логіки та інтерфейсної частини застосунку. Вибір зумовлений універсальністю синтаксису, підтримкою асинхронних операцій, розвиненою екосистемою пакетів та можливістю використання єдиної мови на всіх рівнях архітектури проєкту. Сучасні стандарти мови забезпечують підтримку модульної системи, деструктуризації даних, опціональних ланцюжків викликів

та типізації через розширення, що підвищує надійність коду та спрощує рефакторинг.

Середовище виконання Node.js версії 25.9.0 обрано як платформа для запуску серверної логіки та фонових процесів [47]. Архітектура середовища базується на подієвому циклі та неблокуючому введенні-виведенні, що дозволяє ефективно обробляти паралельні запити до бази даних, виконувати математичні розрахунки сумісності обладнання та генерувати звіти без зупинки інтерфейсу користувача. Оновлена версія середовища містить оптимізації двигуна виконання, покращену роботу з пам'яттю та підтримку сучасних стандартів безпеки, що критично важливо для десктопних застосунків, які працюють з локальними даними користувача. Менеджер пакетів інтегрований у середовище для керування залежностями, фіксації версій бібліотек та автоматичного оновлення компонентів згідно з політикою безпеки.

3.1.4 Фреймворк Electron для кросплатформної розробки десктопних застосунків

Фреймворк Electron реалізує архітектурну основу програми, поєднуючи рушій відображення веб-контенту з середовищем виконання JavaScript для створення нативних десктопних застосунків [44]. Таке рішення дозволяє використовувати стандартні веб-технології для побудови інтерфейсу користувача, зберігаючи при цьому повний доступ до системних API операційної системи. Архітектура фреймворку розділена на головний процес, який керує життєвим циклом вікон, взаємодіє з файловою системою та мережею, та процеси відображення, які відповідають за рендеринг інтерфейсу та обробку дій користувача. Комунікація між процесами реалізована через

захищений канал обміну повідомленнями, що запобігає прямому доступу інтерфейсної частини до критичних системних ресурсів.

Кросплатформність досягається за рахунок уніфікованої оболонки, яка абстрагує відмінності між операційними системами Windows та macOS, забезпечуючи однакову поведінку застосунку на обох платформах. Фреймворк підтримує нативні діалогові вікна вибору файлів, сповіщення системи, керування живленням та інтеграцію з меню операційної системи. Важливим аспектом є підтримка автоматичного оновлення, яке дозволяє доставляти виправлення безпеки та нові функції без необхідності повторного завантаження інсталяційних пакетів користувачем. Безпека реалізована через відключення небезпечних API у процесах відображення, валідацію шляхів до файлів та обмеження прав доступу до локальних ресурсів.

3.1.5 Система управління базами даних SQLite та ORM-бібліотека Prisma

Система управління базами даних SQLite обрана як локальне сховище даних через відсутність потреби у виділеному серверному процесі, нульову конфігурацію розгортання та повну відповідність стандартам SQL. Архітектура системи базується на збереженні всіх даних у єдиному файлі, що спрощує резервне копіювання, перенесення налаштувань між пристроями та забезпечує цілісність інформації навіть у разі аварійного завершення роботи програми. Підтримка транзакцій гарантує атомарність операцій запису, що критично важливо при одночасному оновленні довідників обладнання та збереженні історії розрахунків. Продуктивність системи оптимізована для локальних запитів, а механізми блокування на рівні файлів запобігають пошкодженню даних при паралельному доступі.

ORM-бібліотека Prisma виступає шаром абстракції між прикладним кодом та базою даних, забезпечуючи типобезпечний доступ до таблиць,

автоматичну генерацію міграцій схеми та зручний синтаксис побудови запитів. Використання декларативної мови опису схеми дозволяє візуалізувати структуру бази даних, визначати зв'язки між сутностями та контролювати цілісність даних на етапі проєктування. Бібліотека автоматично генерує клієнтський код з підтримкою автодоповнення, що виключає помилки в назвах полів та типів даних під час написання запитів. Механізм міграцій забезпечує безпечне оновлення структури бази даних при випуску нових версій програми, зберігаючи існуючі записи та адаптуючи таблиці до змінених вимог. Інтеграція з середовищем розробки дозволяє миттєво перевіряти валідність запитів та відстежувати виконання операцій у реальному часі.

Процес перетворення вихідного коду у готовий до використання програмний продукт реалізовано за допомогою спеціалізованих інструментів пакування, які адаптують застосунок під цільові операційні системи. Конфігурація збірки визначає структуру каталогів, включення необхідних бібліотек, оптимізацію розміру виконуваного файлу та налаштування іконок інтерфейсу. Генерація інсталяційних пакетів для Windows реалізована у форматі, що підтримує тиху установку, створення ярликів на робочому столі та реєстрацію протоколів запуску. Для платформи macOS пакет формується з урахуванням вимог до підписування коду, пісочниці виконання та інтеграції з системним меню.

Автоматизація процесу збірки налаштована через конфігураційні файли, які фіксують версії залежностей, параметри стиснення ресурсів та правила виключення відладочних даних. Такий підхід гарантує детермінованість результатів збірки незалежно від робочого середовища розробника та дозволяє відтворювати точні копії програмного продукту для тестування або архівування. Механізм верифікації цілісності пакетів перевіряє контрольні суми файлів перед розповсюдженням, що запобігає поширенню пошкоджених або модифікованих версій застосунку.

Вибір технологічного стеку та інструментів розробки повністю відповідає функціональним та експлуатаційним вимогам програмно-методичного

комплексу. Використання локальної бази даних забезпечує автономність роботи системи без залежності від зовнішніх серверів, що критично важливо для користувачів у регіонах з нестабільним інтернет-з'єднанням та підвищує рівень конфіденційності технічних розрахунків. Кросплатформна архітектура дозволяє охопити максимальну аудиторію користувачів, які працюють на різних операційних системах, не вимагаючи адаптації коду під кожен платформу окремо.

Модульна структура проєкту, реалізована через розділення рівнів відповідальності, забезпечує легкість тестування окремих компонентів, швидке виявлення помилок та можливість заміни окремих модулів без впливу на загальну працездатність системи. Інтеграція системи контролю версій та автоматизованих конвеєрів збірки мінімізує ризик людського фактора під час випуску оновлень та гарантує стабільність продуктивної версії програми. Обрані технології підтримують масштабування функціоналу, що дозволяє у майбутньому додавати нові модулі аналітики, розширені метеорологічні розрахунки або інтеграцію з зовнішніми каталогами обладнання без зміни базової архітектури.

Таким чином, технологічний стек та середовище розробки обрано на підставі комплексного аналізу вимог до продуктивності, безпеки, кросплатформності та зручності супроводу. Реалізовані рішення забезпечують стабільну роботу програмного комплексу, ефективне керування кодовою базою та готовність до подальшого розвитку відповідно до потреб ринку альтернативної енергетики.

3.2 Проєктування структури програмно-методичного комплексу

Проєктування архітектури програмного забезпечення є критичним етапом життєвого циклу розробки, який визначає масштабованість, надійність,

зручність супроводу та відповідність системи вимогам замовника [43]. Для програмно-методичного комплексу підбору сонячних панелей та інверторів було обрано багат шарову архітектуру з чітким поділом відповідальності між рівнями представлення, бізнес-логіки, доступу до даних та сховища інформації. Такий підхід дозволяє ізолювати зміни в інтерфейсі від змін у алгоритмах розрахунків, спрощує тестування окремих компонентів та забезпечує можливість подальшого розширення функціоналу без необхідності перебудови всієї системи.

3.2.1 Багат шарова архітектура та принципи поділу відповідальності

Архітектурна модель комплексу базується на чотирирівневій структурі, яка адаптована до особливостей десктопного середовища Electron та середовища виконання Node.js. Кожен рівень виконує визначену функцію та виключає пряме звернення між несуміжними рівнями [55].

Рівень представлення відповідає за взаємодію з користувачем, відображення вхідних форм, виведення результатів розрахунків, формування звітів та управління станом інтерфейсу. Цей рівень реалізовано у процесах відображення Electron, які виконуються в ізолюваному контексті з обмеженим доступом до системних API. Рівень представлення не містить бізнес-правил та не виконує прямих запитів до бази даних, що запобігає порушенню цілісності даних та знижує ризик несанкціонованого доступу.

Рівень бізнес-логіки інкапсулює основні алгоритми підбору обладнання, перевірки технічної сумісності, валідації конфігурацій масиву та формування рекомендацій. Цей шар виконується в головному процесі Electron або у виділених фонових потоках, що дозволяє уникнути блокування інтерфейсу під час виконання розрахунків. Бізнес-логіка отримує валідовані параметри від рівня представлення, звертається до рівня доступу до даних для отримання

характеристик обладнання, виконує математичні перевірки та повертає структурований результат.

Рівень доступу до даних забезпечує абстракцію між бізнес-логікою та фізичним сховищем. Він реалізує операції створення, читання, оновлення та видалення записів, керує транзакціями, контролює цілісність посилань та оптимізує запити до бази даних. У проекті цей рівень реалізовано через ORM-бібліотеку Prisma, яка генерує типобезпечний клієнтський код на основі декларативної схеми [48], [49]. Такий підхід виключає синтаксичні помилки в SQL-запитах, автоматично контролює типи даних та спрощує міграції структури бази при оновленні версії програми.

Рівень сховища даних представлений реляційною системою управління базами даних SQLite, яка зберігає всю інформацію в єдиному локальному файлі. Цей рівень забезпечує атомарність транзакцій, підтримку зовнішніх ключів, індексацію для прискорення пошуку та механізми блокування на рівні файлів для запобігання пошкодженню даних при паралельному доступі.

Принцип поділу відповідальності реалізовано через чітке розмежування функціональних зон. Модулі рівня представлення відповідають лише за відображення та збір даних, модулі бізнес-логіки виконують розрахунки та прийняття рішень, модулі доступу до даних керують збереженням, а рівень сховища забезпечує фізичне зберігання. Така ізоляція дозволяє замінювати окремі компоненти без впливу на інші шари, спрощує модульне тестування та забезпечує стабільність системи під час впровадження нових функцій.

3.2.2 Організація кодової бази та структура каталогів проекту

Організація файлової структури проекту виконана з урахуванням вимог до модульності, зручності навігації та стандартизації шляхів до ресурсів. Кодова база поділена на логічні директорії, кожна з яких відповідає

визначеному шару архітектури або допоміжній підсистемі. Така структура сприяє швидкому пошуку необхідних файлів, полегшує інтеграцію нових розробників у проєкт та забезпечує однорідність підходу до іменування компонентів. Структура проєкту представлена на рисунку 3.2.

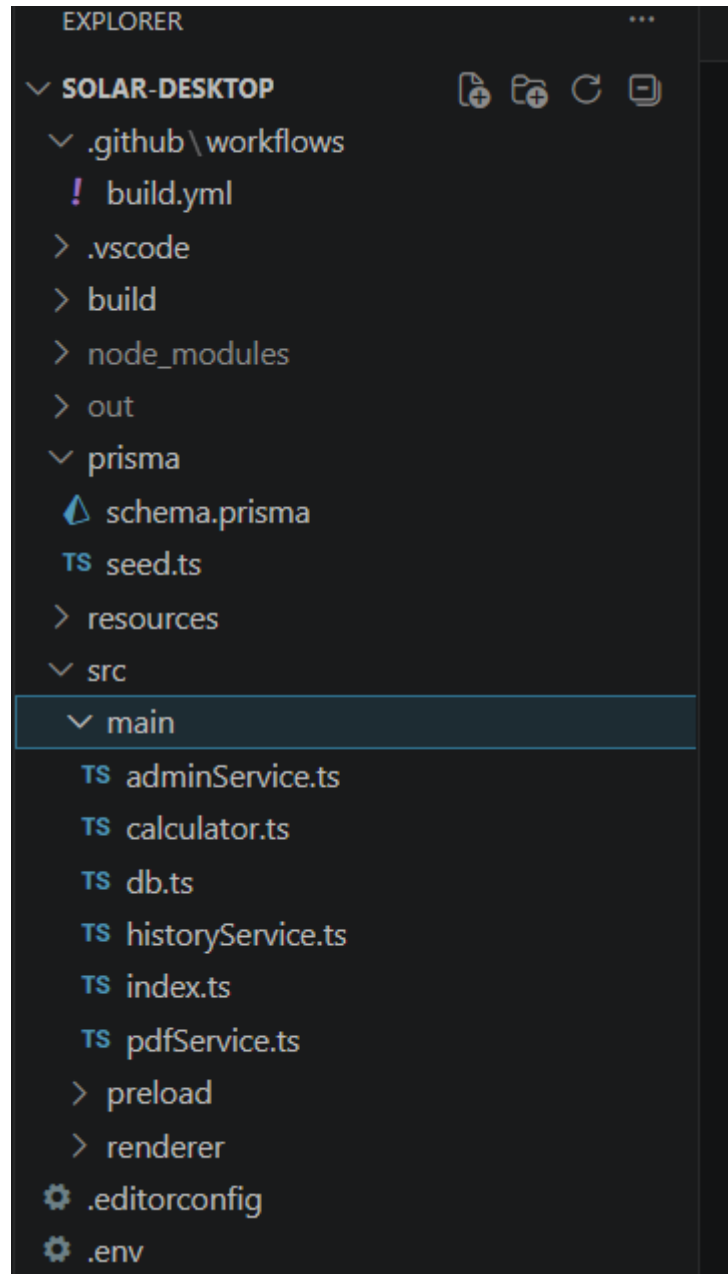


Рисунок 3.2 – Організація файлової структури проєкту

Кореневий каталог містить конфігураційні файли середовища розробки, маніфест залежностей, налаштування збірки та скрипти автоматизації.

Директорія вихідного коду поділена на підкаталоги, що відповідають процесам Electron та рівням бізнес-логіки. Головний процес містить модулі ініціалізації вікон, обробники міжпроцесних викликів, керування життєвим циклом програми та ініціалізацію підключення до бази даних. Процеси відображення містять компоненти інтерфейсу, обробники подій користувача, валідатори вхідних даних та модулі формування звітів.

Каталог бізнес-логіки згруповано за функціональними модулями: підбір обладнання, перевірка сумісності, валідація конфігурацій, управління довідниками та експорт результатів. Кожен модуль реалізовано у вигляді окремих файлів або класів, що містять ізольовані методи та не залежать від конкретної реалізації інтерфейсу. Рівень доступу до даних винесено в окрему директорію, де зберігаються файли схеми бази даних, міграції, генерований клієнтський код ORM та допоміжні функції для роботи з транзакціями.

Конфігураційні дані, такі як шляхи до ресурсів, параметри збірки, змінні середовища та константи розрахунків, винесено в окремі модулі, що імпортується за потреби. Такий підхід дозволяє змінювати налаштування без втручання в основну логіку програми та забезпечує гнучкість під час розгортання на різних операційних системах. Додатково реалізовано механізм валідації конфігурації при запуску програми, що перевіряє наявність обов'язкових файлів, коректність шляхів та доступність довідників обладнання.

3.2.3 Механізми міжмодульної взаємодії та керування конфігурацією

Взаємодія між рівнями представлення та бізнес-логіки реалізована через механізм міжпроцесної комунікації, що забезпечує безпечний обмін даними між ізольованими процесами Electron [61]. Інтерфейс користувача надсилає запити через асинхронні канали, які обробляються в головному

процесі, де виконуються розрахунки та звернення до бази даних. Результати повертаються у структурованому вигляді, що дозволяє інтерфейсу оновлювати стан без блокування основного потоку. Такий підхід виключає прямий доступ рівня представлення до системних ресурсів та забезпечує дотримання політики безпеки контексту ізоляції.

Керування конфігурацією програми виконано через централізований модуль, який завантажує параметри з файлів середовища, перевіряє їх валідність та ініціалізує глобальні константи. Змінні середовища використовуються для зберігання шляхів до локальної бази даних, параметрів логування, налаштувань збірки та прапорців режиму розробки. При запуску програми виконується перевірка наявності обов'язкових конфігурацій, а у разі відсутності критичних параметрів система ініціює процес відновлення або повідомляє користувача про помилку ініціалізації.

Для забезпечення стабільності роботи реалізовано механізм обробки виключень на кожному рівні взаємодії. Помилки валідації вхідних даних перехоплюються на рівні представлення та виводяться у зручному для користувача форматі. Помилки розрахунків або звернень до бази даних логуються в системний журнал, а користувачу надається загальне повідомлення про неможливість виконання операції без розкриття технічних деталей. Такий підхід забезпечує захист внутрішньої логіки програми та запобігає витоку конфіденційної інформації у разі збоїв.

3.2.4 Проектування реляційної моделі даних та ER-діаграма

Реляційна модель даних програмно-методичного комплексу розроблена з урахуванням принципів нормалізації, цілісності посилань та оптимізації запитів для локального середовища виконання. Структура бази даних включає чотири основні сутності, які відображають ключові об'єкти

предметної області: користувачі системи, каталог інверторів, каталог сонячних панелей та журнал виконаних розрахунків. Кожна сутність містить набір атрибутів, що описують її стан, та індекси для прискорення пошуку за ключовими полями. На рисунку 3.3 представлена ER діаграма спроектованої бази даних.

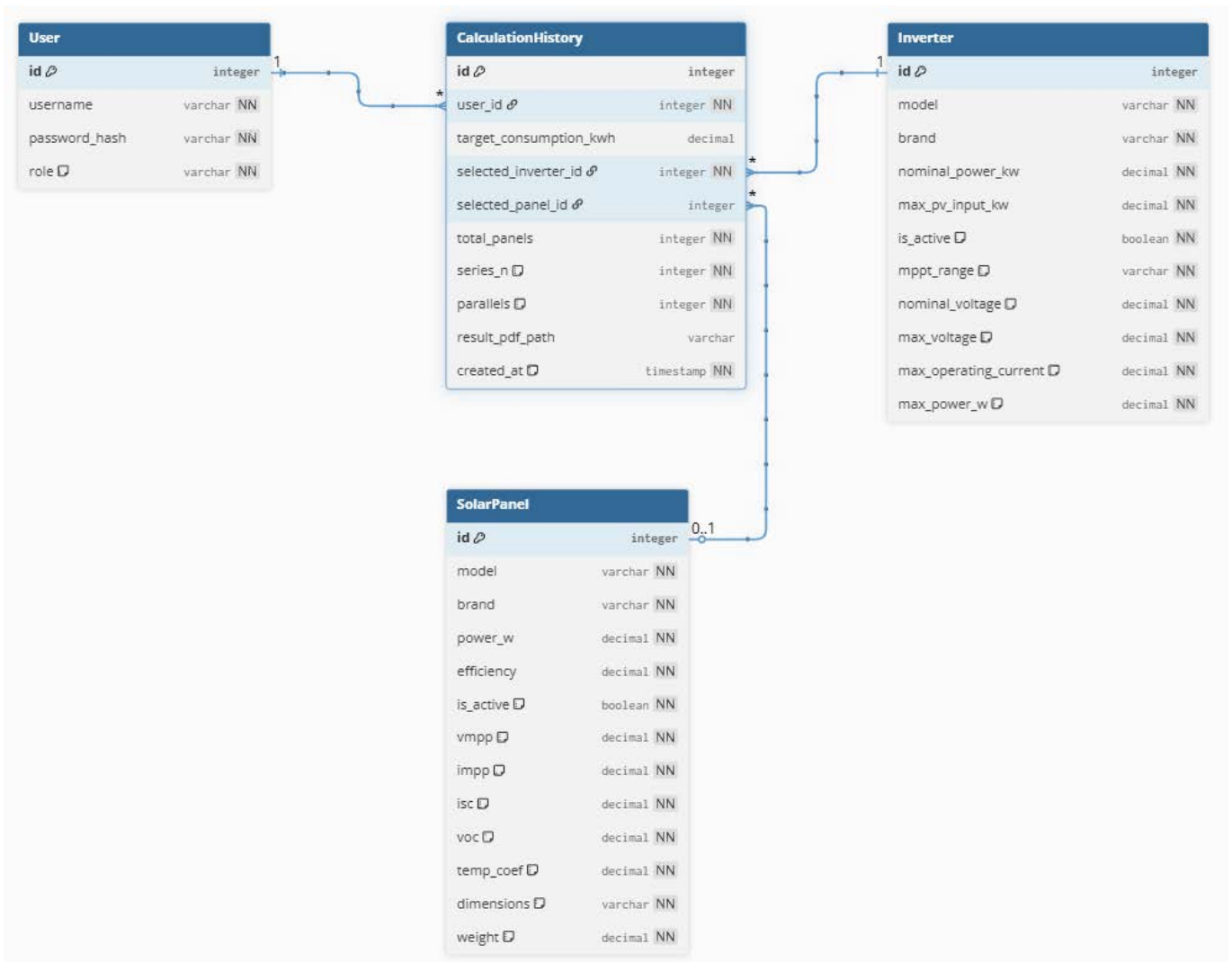


Рисунок 3.3 – ER діаграма бази даних

Сутність користувачів призначена для реалізації механізму авторизації та розподілу прав доступу. Вона містить ідентифікатор, ім'я користувача, хеш пароля та роль, яка визначає доступ до функцій адміністрування каталогів обладнання.

Сутність інверторів зберігає технічні характеристики перетворювачів струму: модель, виробника, номінальну потужність, діапазон вхідної напруги, максимальний робочий струм, максимальну потужність та прапорець активності.

Сутність сонячних панелей містить дані про фотоелектричні модулі: модель, виробника, потужність, коефіцієнт корисної дії, напругу та струм у точці максимальної потужності, напругу холостого ходу, струм короткого замикання, температурний коефіцієнт, габарити та вагу.

Сутність історії розрахунків фіксує результати виконаних підборів обладнання. Вона містить ідентифікатор розрахунку, посилання на користувача, ідентифікатори обраного інвертора та панелі, параметри конфігурації масиву, загальну кількість модулів, шлях до згенерованого звіту та часову мітку створення запису. Зовнішні ключі забезпечують зв'язок між розрахунками та довідниками обладнання, а також контролюють цілісність даних при видаленні записів. Для історії розрахунків застосовано каскадне видалення посилань на користувача та обмеження на видалення інверторів і панелей, щоб уникнути втрати архівних даних.

Проектована структура бази даних відповідає третій нормальній формі, що виключає транзитивні залежності, мінімізує дублювання даних та забезпечує узгодженість інформації при оновленні довідників. Індокси створено на полях, які використовуються для фільтрації та пошуку: модель обладнання, виробник, прапорець активності та ідентифікатори у зв'язках. Такий підхід гарантує швидкий доступ до даних навіть при зростанні обсягу каталогів та історії розрахунків.

3.2.5 Об'єктно-орієнтоване моделювання системи та UML-діаграма класів

Об'єктно-орієнтоване моделювання програмно-методичного комплексу виконано з метою формалізації взаємодії між компонентами системи, визначення атрибутів та методів класів, а також встановлення зв'язків між сутностями на рівні програмного коду [42], [60].

Діаграма класів відображає статичну структуру системи, яка реалізована у вихідному коді проекту, та слугує основою для побудови логіки взаємодії між рівнями архітектури. На рисунку 3.4 наведена UML-діаграма класів розробленого проекту.

Основний клас інвертора інкапсулює технічні характеристики перетворювача струму та методи валідації вхідних параметрів. Він містить атрибути ідентифікатора, моделі, виробника, номінальної потужності, максимальної вхідної напруги, максимального робочого струму та діапазону відстеження точки максимальної потужності. Клас реалізує методи перевірки відповідності напруги та струму масиву панелей, розрахунку допустимої потужності навантаження та формування технічної специфікації для звіту.

Клас сонячної панелі зберігає параметри фотоелектричного модуля та алгоритми розрахунку системних характеристик масиву. Атрибути включають ідентифікатор, модель, виробника, номінальну потужність, коефіцієнт корисної дії, напругу та струм у точці максимальної потужності, напругу холостого ходу, струм короткого замикання та температурний коефіцієнт. Методи класу виконують розрахунок загальної напруги стрінга, загального струму паралельних гілок, перевірку перевищення граничних значень інвертора та формування пояснювального повідомлення у разі невідповідності.

Клас конфігурації масиву агрегує параметри підключення панелей та керує валідацією схеми з'єднання. Він містить атрибути кількості модулів у стрінзі, кількості паралельних гілок, загальної кількості панелей та посилання

на обрані сутності інвертора та панелі. Методи класу перевіряють коректність вхідних даних, розраховують системні параметри, викликають методи перевірки сумісності та повертають структурований результат підбору.

Клас розрахункового двигуна інкапсулює бізнес-логіку підбору обладнання та керує процесом перебору сумісних конфігурацій. Він містить методи ініціалізації розрахунків, фільтрації каталогу панелей за виробником, виконання перевірок для кожного модуля, сортування результатів за ознакою сумісності та потужністю, а також формування підсумкового звіту. Клас не залежить від конкретної реалізації інтерфейсу та працює виключно з об'єктами предметної області.

Клас сервісу бази даних забезпечує абстракцію доступу до сховища інформації та керує транзакціями запису. Він містить методи підключення до SQLite, виконання запитів через Prisma Client, збереження історії розрахунків, отримання списків обладнання, оновлення довідників адміністратором та обробку помилок цілісності даних. Клас реалізує патерн репозиторій, що дозволяє замінювати реалізацію сховища без зміни логіки бізнес-шару.

Клас контролера інтерфейсу керує взаємодією з користувачем, обробляє події введення даних, викликає методи розрахункового двигуна та оновлює стан відображення. Він містить методи валідації вхідних параметрів, ініціалізації запитів до бізнес-логіки, обробки відповідей, формування модальних вікон характеристик панелей та експорту звітів у PDF. Клас не містить бізнес-правил та виконує лише функції посередника між рівнями представлення та логіки.

Зв'язки між класами реалізовано через асоціації, агрегації та залежності. Клас конфігурації масиву агрегує об'єкти інвертора та панелі, оскільки не може існувати без вибору конкретних моделей. Розрахунковий двигун залежить від класів панелі та інвертора для виконання перевірок, а сервіс бази даних залежить від структури схеми Prisma для генерації запитів. Контролер інтерфейсу залежить від розрахункового двигуна та сервісу бази даних, але

не має прямого зв'язку з ними, що забезпечує слабку зв'язаність та високу модульність системи.

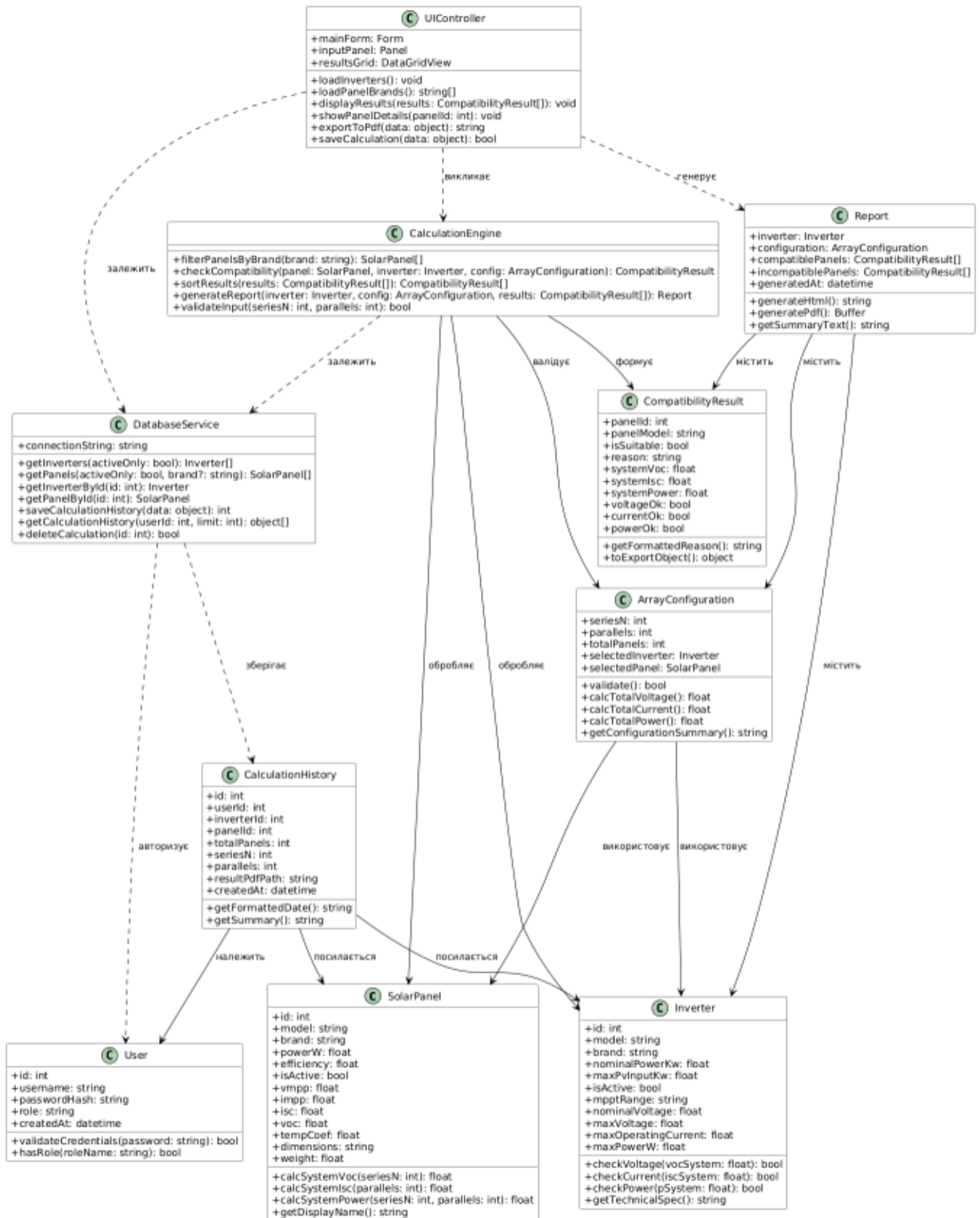


Рисунок 3.4 – UML-діаграма класів

3.3 Реалізація програмного комплексу та інтерфейсу користувача

Архітектура застосунку базується на розділенні відповідальності між головним backend процесом Electron та процесом відображення frontend. Головний процес відповідає за взаємодію з файловою системою, базою даних SQLite через ORM Prisma, генерацію PDF-звітів та системні операції. Процес відображення відповідає за графічний інтерфейс користувача, валідацію вхідних даних та відображення результатів розрахунків.

Графічний інтерфейс застосунку «Solar System Calculator» розроблявся з урахуванням сучасних стандартів UX/UI-дизайну та вимог до ергономіки робочого місця інженерів-проектувальників та менеджерів з продажу сонячного енергетичного обладнання. Основною метою проектування інтерфейсу було створення максимально зрозумілого, логічно структурованого та адаптивного середовища, яке дозволяє користувачеві швидко виконувати підбір обладнання, аналізувати сумісність фотомодулів та інверторів, а також керувати базою даних.

Інтерфейс застосунку реалізовано за допомогою бібліотеки React та фреймворку Tailwind CSS, що забезпечує гнучку стилізацію компонентів та плавні переходи між станами [50], [51]. Весь інтерфейс розділено на два основних режими роботи:

- Панель користувача – призначена для підбору обладнання, перевірки його сумісності, перегляду історії розрахунків та експорту результатів у форматі PDF.
- Панель адміністратора – призначена для додавання, видалення, редагування та тимчасового приховування технічних характеристик обладнання у базі даних SQLite.

Нижче наведено детальний опис кожної екранної форми застосунку, опис інтерактивних елементів та відповідні фрагменти коду, що забезпечують їхнє функціонування.

3.3.1 Головне вікно калькулятора

Головне вікно програми є інтерактивною робочою областю розрахунків. Зовнішній вигляд цієї панелі представлено на рисунку 3.5

Вхідні дані

Інвертор
Оберіть інвертор...

Фільтр панелей (Бренд)
Всі бренди

N у серії
0

Кількість рядків (паралель)
0

Підібрати обладнання

Результат перевірки

Введіть дані та натисніть кнопку для перевірки

Історія перевірок (останні 10)

ДАТА	ІНВЕРТОР	ПАНЕЛЬ	СХЕМА (N x ПАР)
09.06.2026	PowMr POW-HVM3.2H-24V	Trina Solar Vertex S+ NEG9R.28-430W	4 x 1

Рисунок 3.5 – Головне вікно застосунку

Вікно візуально розділено на три основні функціональні зони:

1. Верхня панель (Header) – містить назву застосунку «Solar System Calculator», опис його призначення та кнопку «Панель Адміністратора» для швидкого переходу до панелі авторизації та керування базою даних.

2. Ліва колонка (Форма «Вхідні дані») – містить елементи управління для введення вихідних параметрів проектованої системи:

- Випадаючий список «Інвертор» – завантажує з бази даних та відображає список усіх активних інверторів із зазначенням бренду, моделі та номінальної АС-потужності у кВт.

- Випадаючий список «Фільтр панелей (Бренд)» – дозволяє відфільтрувати список сонячних панелей за конкретним виробником (або відобразити всі бренди).

- Поле введення «N у серії» – задає кількість фотомодулів, які підключаються послідовно в один стрінг (за замовчуванням встановлено значення 10).

- Поле введення «Кількість рядків (паралель)» – задає кількість паралельних стрінгів, що підключаються до інвертора (за замовчуванням встановлено значення 1).

- Кнопка «Підібрати обладнання» – ініціює процес розрахунку сумісності. Під час обробки запиту кнопка змінює текст на «Перевірка...» та блокується для запобігання повторним клікам.

3. Права колонка (Секція «Результат перевірки») – динамічна область, яка до виконання розрахунку відображає текстову підказку «Введіть дані та натисніть кнопку для перевірки». Після натискання кнопки відображає:

- Інформаційну плашку із ключовими параметрами обраного інвертора (діапазон MPPT, номінальна напруга, максимальний струм, максимальна потужність) та схемою підключення.

- Таблицю сумісних та несумісних сонячних панелей із колірним кодуванням результатів.

Логіка обробки події відправки форми та взаємодія з основним процесом Electron через IPC-міст (Inter-Process Communication) представлена у фрагменті коду на рисунку 3.6.

```

const handleSubmit = async (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault();
  const formData = new FormData(e.currentTarget);
  const inverterId = Number(formData.get('inverterId'));
  const brand = formData.get('brand') as string;
  const seriesN = Number(formData.get('seriesN'));
  const parallels = Number(formData.get('parallels'));

  if (!inverterId || !seriesN || !parallels) return;

  dispatch(setLoading(true));
  setSelectedResultIndex(null);
  try {
    // Виклик IPC-методу розрахунку в головному процесі
    const calcResult = await window.api.calculateSystem({
      inverterId,
      seriesN,
      parallels,
      panelBrand: brand
    });
    dispatch(setCalculationResults(calcResult));
  } catch (error: any) {
    console.error("Calculation error:", error);
    alert(error.message || "Сталася помилка під час розрахунку");
  } finally {
    dispatch(setLoading(false));
  }
};

```

Рисунок 3.6 – Обробка форми вхідних даних
та запуск розрахунку сумісності

Таблиця результатів розрахунку підтримує інтерактивний вибір оптимального варіанту через перемикачі `radio`. Кожен рядок таблиці забарвлюється залежно від статусу сумісності:

- Зелений колір (`bg-green-50` або `hover:bg-green-50`) – фотомодуль повністю сумісний із вибраним інвертором за напругою, струмом та сумарною

потужністю. У стовпці «Причина допуску / Помилка» відображається статус «Відповідає характеристикам обладнання».

– Червоний колір (`bg-red-50 opacity-75`) – обладнання несумісне. Перемикач вибору блокується (`disabled`), а в стовпці помилки детально вказується причина відхилення, наприклад перевищення ліміту напруги холостого ходу чи максимального робочого струму.

Після вибору конкретної сумісної моделі панелі активується кнопка «Зберегти вибір та згенерувати PDF». Ця кнопка ініціює запис результату розрахунку в історію бази даних та формує детальний звіт, який експортується у PDF за допомогою головного процесу Electron що наведено у фрагменті коду на рисунку 3.7.

```
const handleSaveAndExport = async () => {
  if (selectedResultIndex === null || !calculationData) return;
  const selectedResult = calculationData.results[selectedResultIndex];
  const inverter = calculationData.inverter;
  const totalPanels = calculationData.seriesN * calculationData.parallels;

  // 1. Запис розрахунку до бази даних
  try {
    await window.api.saveCalculation({
      userId: 1,
      inverterId: inverter.id,
      panelId: selectedResult.panel.id,
      totalPanels: totalPanels,
      seriesN: calculationData.seriesN,
      parallels: calculationData.parallels
    });
    const hist = await window.api.getHistory();
    dispatch(setHistory(hist));
  } catch (e) {
    console.error("Save error:", e);
  }
}
```

Рисунок 3.7 – Збереження вибору обладнання та експорт PDF-звіту

Генерація HTML-шаблону для PDF звіту [52] та виклик сервісу експорту PDF у головному процесі наведено на рисунку 3.8.

```
const htmlString = `
<html>
  <head>
    <style>
      body { font-family: sans-serif; padding: 40px; color: #333; }
      h1 { color: #1e3a8a; border-bottom: 2px solid #e5e7eb; padding-bottom: 10px; }
      .box { background: #f3f4f6; padding: 15px; margin-bottom: 15px; border-radius: 8px; }
      table { width: 100%; border-collapse: collapse; margin-top: 20px; font-size: 14px; }
      th, td { border: 1px solid #ddd; padding: 8px; text-align: left; }
      th { background-color: #f8f9fa; }
      .ok { color: green; font-weight: bold; }
    </style>
  </head>
  <body>
    <h1>Звіт про підбір обладнання СЕС</h1>
    <div class="box">
      <p><strong>Інвертор:</strong> ${inverter.brand} ${inverter.model}, AC Потужність: ${inverter.nominal_power_kw}
      кВт</p>
      <p><strong>Підсумок:</strong> MPPT-1 (${selectedResult.p_system_w.toFixed(0)} Вт) / ліміт ${inverter.
      max_power_w} Вт - <span class="ok">ОК</span>.</p>
    </div>
    ...
  </body>
</html>
`;

// 3. Виклик сервісу експорту PDF у головному процесі
try {
  const filePath = await window.api.exportPdf(htmlString);
  if (filePath) alert(`Звіт збережено: ${filePath}`);
} catch (e) {
  alert("Помилка при збереженні PDF");
}
};
```

Рисунок 3.8 – Генерація HTML-шаблону для PDF звіту

Зовнішній вигляд вікна збереження результату підбору представлено на рисунку 3.9

Інвертор
Must PV18-5548 VHM (5.5кВт)

Фільтр панелей (Бренд)
Всі бренди

N у серії
3

Кількість рядків (паралель)
2

Підібрати обладнання

Обрано: Must PV18-5548 VHM | **Діапазон MPPT:** 60-130В | **Номінальна напруга:** 90 В |
Максимальний струм: 80 А | **Максимальна потужність:** 5500 Вт |
Схема підключення: 3 послідовно × 2 паралельно

<input type="radio"/>	Canadian Solar HiKu6 CS6R-MS-395W	2.37 кВт	Відповідає характеристикам обладнання
<input checked="" type="radio"/>	Risen Energy Titan S RSM40-8-395W	2.37 кВт	Відповідає характеристикам обладнання
<input type="radio"/>	Longi Solar LR5-72HTH-585W	3.51 кВт	Перевищено напругу (Voc 156.6В > 145В)
<input type="radio"/>	Jinko Solar Tiger Neo N-type 72HL4-585W	3.51 кВт	Перевищено напругу (Voc 152.9В > 145В)
<input type="radio"/>	Longi Solar LR5-72HTH-580W	3.48 кВт	Перевищено напругу (Voc 156.3В > 145В)
<input type="radio"/>	Jinko Solar Tiger Neo N-type 72HL4-580W	3.48 кВт	Перевищено напругу (Voc 152.6В > 145В)
<input type="radio"/>	Jinko Solar Tiger Neo N-type		Перевищено напругу (Voc

Зберегти вибір та згенерувати PDF

Історія перевірок (останні 10)

ДАТА	ІНВЕРТОР	ПАНЕЛЬ	СХЕМА (N × ПАР)
15.06.2026	Must PV18-5548 VHM	Risen Energy Titan S RSM40-8-395W	3 × 2
15.06.2026	Victron EasySolar-II 48/3000/35-32	Jinko Solar Tiger Pro 54HC-395W	5 × 1

Рисунок 3.9 – Вікно збереження результатів підбору

3.3.2 Модальне вікно характеристик фотомодуля

При натисканні на назву сонячної панелі в таблиці результатів користувачеві відкривається модальне вікно з детальними технічними характеристиками вибраного пристрою. Зовнішній вигляд вікна представлено на рисунку 3.10.

Jinko Solar Tiger Neo N-type 72HL4-585W	
Потужність	585 Вт
Напруга при макс. потужності (V_{mpp})	42.27 В
Струм при макс. потужності (I_{mpp})	13.84 А
Струм короткого замикання (I_{sc})	14.81 А
Напруга холостого ходу (V_{oc})	50.97 В
ККД фотомодуля	22.2%
Температурний коефіцієнт V_{oc}	-0.25%/°C
Габарити (A×L×T)	2278x1134x35
Вага	28 кг

[Закрити](#)

Рисунок 3.10 – Модальне вікно характеристик фотомодуля

Це вікно виконує роль віртуального паспорта обладнання та містить такі поля:

- Номінальна потужність (Вт);
- Напруга в точці максимальної потужності (V_{mpp} , В);
- Струм у точці максимальної потужності (I_{mpp} , А);
- Струм короткого замикання (I_{sc} , А);
- Напруга холостого ходу (V_{oc} , В);
- Коефіцієнт корисної дії фотомодуля (ККД, %);
- Температурний коефіцієнт напруги холостого ходу (%/°C);
- Габаритні розміри (довжина, ширина, товщина в мм);
- Фізична вага пристрою (кг).

Завдяки цьому інженер може оперативно звірити фізичні габарити панелей для проектування опорних металоконструкцій на об'єкті.

3.3.3 Історія перевірок

У нижній частині головного вікна розташовано таблицю останніх 10 перевірок підбору обладнання зображену на рисунку 3.11. Вона завантажується автоматично при запуску застосунку та оновлюється після кожного успішного збереження розрахунку.

Історія перевірок (останні 10)			
ДАТА	ІНВЕРТОР	ПАНЕЛЬ	СХЕМА (N × ПАР)
13.06.2026	Victron EasySolar-II 48/3000/35-32	Jinko Solar Tiger Pro 72HC-540W	5 × 1
13.06.2026	Deye SUN-8K-SG01LP1	Trina Solar Vertex S+ NEG9R.28-440W	4 × 2
13.06.2026	Deye SUN-8K-SG01LP1	Risen Energy Titan S RSM40-8-400W	4 × 2
13.06.2026	Must PV18-3024 VHM	Risen Energy Hyper-ion RSM132-8-690W	2 × 2
13.06.2026	Must PV18-3024 VHM	Risen Energy Hyper-ion RSM132-8-710W	2 × 2
09.06.2026	PowMr POW-HVM3.2H-24V	Trina Solar Vertex S+ NEG9R.28-430W	4 × 1

Рисунок 3.11 – Секція історії перевірок та підбору обладнання

Історія підбору у вигляді таблиці містить стовпці:

- Дата – дата збереження розрахунку.
- Інвертор – марка та модель інвертора.
- Панель – марка та модель підібраної сонячної панелі.
- Схема (N × Пар) – конфігурація підключення масиву фотомодулів (послідовно × паралельно).

3.3.4 Панель авторизації адміністратора

Для запобігання несанкціонованій зміні бази даних обладнання, доступ до адмін-панелі захищено паролем. Форму входу представлено на рисунку 3.12.

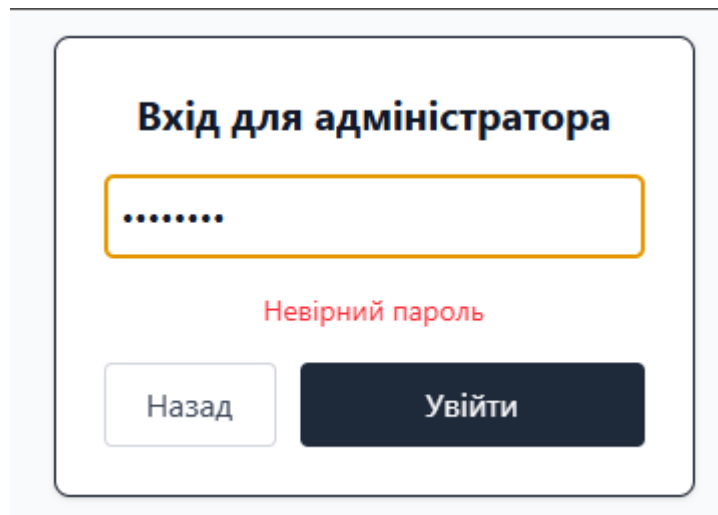


Рисунок 3.12 – Вікно авторизації адміністратора

Якщо пароль введено неправильно, під полем введення відображається повідомлення про помилку червоним кольором («Невірний пароль»), а кнопка входу залишається активною для повторних спроб. Обробка авторизації реалізована через безпечну верифікацію на рівні сервісів Node.js головного процесу Electron, що виключає зберігання пароля у вихідному коді клієнтської частини. Перевірка автентифікації виконується через IPC-виклик `window.api.verifyAdmin(password)`, який делегує завдання головному процесу Electron. Фрагмент коду перевірки автентифікації адміністратора наведено на рисунку 3.13.

```

26   const handleLogin = async (e: React.FormEvent) => {
27     e.preventDefault();
28     const isValid = await window.api.verifyAdmin(password);
29     if (isValid) { dispatchAdmin(setAuth(true)); setError(''); }
30     else { setError('Невірний пароль'); }
31   };

```

Рисунок 3.13 – Фрагмент коду перевірки автентифікації адміністратора

3.3.5 Панель адміністратора

Після успішної авторизації користувачеві відкривається інтерфейс адміністратора наведений на рисунку 3.14.



The screenshot shows a web application window titled "Electron". The main content area is titled "Панель керування обладнанням" (Equipment Management Panel). In the top right corner, there is a link "Вийти до калькулятора" (Go to calculator). Below the title, there is a section for "Інвертори" (Inverters) with a blue button "+ Додати інвертор" (+ Add inverter). The main part of the interface is a table with the following columns: "БРЕНД" (Brand), "МОДЕЛЬ" (Model), "ПОТУЖНІСТЬ (кВт)" (Power (kW)), "СТАТУС" (Status), and "ДІЇ" (Actions). The table lists several inverters, all with a status of "Активний" (Active).

БРЕНД	МОДЕЛЬ	ПОТУЖНІСТЬ (кВт)	СТАТУС	ДІЇ
Victron	MultiPlus-II 48/8000 (w/ MPPT RS)	6.4	Активний	👁️ ✎️ 🗑️
Victron	Inverter RS Smart 48/6000	5.3	Активний	👁️ ✎️ 🗑️
Victron	Multi RS Solar 48/6000	5.3	Активний	👁️ ✎️ 🗑️
Victron	EasySolar-II 48/5000/70-50	4	Активний	👁️ ✎️ 🗑️
Victron	EasySolar-II 48/3000/35-32	2.4	Активний	👁️ ✎️ 🗑️
PowMr	POW-SunSmart 10K	10	Активний	👁️ ✎️ 🗑️
PowMr	POW-SunSmart 8K	8	Активний	👁️ ✎️ 🗑️
PowMr	POW-SunSmart 5.5K	5.5	Активний	👁️ ✎️ 🗑️

Рисунок 3.14 – Інтерфейс адміністратора

Панель розділена на дві аналогічні секції: «Інвертори» та «Сонячні панелі». Зовнішній вигляд операційного вікна керування обладнанням наведено на рисунку 3.15.

Deye	SUN-3K-SG04LP1	3	Активний	👁️ 📄 🗑️
------	----------------	---	----------	---------

Сонячні панелі				+ Додати панель
БРЕНД	МОДЕЛЬ	ПОТУЖНІСТЬ (Вт)	СТАТУС	ДІЇ
Canadian Solar	CS6.1-54TD-470W	470	Активна	👁️ 📄 🗑️
Canadian Solar	CS6.1-54TD-465W	465	Прихована	👁️ 📄 🗑️
Canadian Solar	CS6.1-54TD-460W	460	Активна	👁️ 📄 🗑️
Canadian Solar	CS6.1-54TD-455W	455	Активна	👁️ 📄 🗑️
Canadian Solar	CS6.1-54TD-450W	450	Активна	👁️ 📄 🗑️

Рисунок 3.15 – Операційне вікно керування обладнанням

У кожній секції відображається таблиця наявного обладнання з можливістю виконання наведених нижче операцій.

Додавання нового обладнання здійснюється за допомогою кнопок «+ Додати інвертор» та «+ Додати панель». Форму додавання/редагування інвертора представлено на рисунку 3.16.

Наведена форма містить поля для введення:

- Бренд та Модель пристрою (текстові поля);
- Номінальна потужність АС (кВт) та Макс. потужність PV (кВт);
- Діапазон напруг MPPT (В) (текстовий діапазон, наприклад, «120-430»);
- Номінальна напруга (В) та Максимально допустима напруга (В);
- Максимальний робочий струм інвертора (А) та Максимальна потужність (Вт).

Додати Інвертор ×

Бренд

Модель

Номінальна потужність АС (кВт)

Макс. потужність PV (кВт)

Діапазон MPPT (напр. 120-430)

Номінальна напруга (В)

Макс. напруга (В)

Макс. робочий струм (А)

Макс. потужність (Вт)

Рисунок 3.16 – Модальне вікно додавання/редагування інвертора

Форму додавання/редагування сонячної панелі представлено на рисунку 3.17.

The image shows a modal window titled "Додати Панель" (Add Panel) with a close button (X) in the top right corner. The window contains several input fields for solar panel specifications, arranged in two columns. The fields are:










- Бренд (Brand)
- Модель (Model)
- Потужність (Вт) (Power (W))
- ККД (%) (Efficiency (%))
- Напруга V_{mp} (В) (Voltage V_{mp} (V))
- Струм I_{mp} (А) (Current I_{mp} (A))
- Струм КЗ I_{sc} (А) (Short-circuit current I_{sc} (A))
- Напруга ХХ V_{oc} (В) (Open-circuit voltage V_{oc} (V))
- Темп. коеф. V_{oc} (%/°C) (Temperature coefficient V_{oc} (%/°C))
- Габарити (напр. 1722x1134x30) (Dimensions (e.g., 1722x1134x30))
- Вага (кг) (Weight (kg))

At the bottom right of the modal window, there are two buttons: "Скасувати" (Cancel) and "Зберегти" (Save).

Рисунок 3.17 – Модальне вікно додавання/редагування сонячної панелі

Всі числові поля мають встановлений атрибут `type="number"` з відповідними кроками регулювання (`step="0.1"` чи `step="0.01"`), що забезпечує вбудовану HTML5 валідацію значень на рівні браузера перед надсиланням даних. Кнопка «Скасувати» дозволяє вийти з режиму редагування без збереження змін, а кнопка «Зберегти» фіксує нові характеристики в базі даних.

Активация / деактивация (приховування) пристрою здійснюється натисканням на іконку «око» (інтерактивні кнопки «EyeIcon» та «EyeOffIcon»). Приклад кнопки наведено на рисунку 3.18.

Deye	SUN-5K-SG03LP1	5	Активний	  
Deye	SUN-3.6K-SG03LP1	3.6	Прихований	  
Deye	SUN-3K-SG04LP1	3	Активний	  
















Сонячні панелі					+ Додати панель
БРЕНД	МОДЕЛЬ	ПОТУЖНІСТЬ (Вт)	СТАТУС	Дії	
Canadian Solar	CS6.1-54TD-470W	470	Активна	  	
Canadian Solar	CS6.1-54TD-465W	465	Прихована	  	
Canadian Solar	CS6.1-54TD-460W	460	Активна	  	
Canadian Solar	CS6.1-54TD-455W	455	Прихована	  	
Canadian Solar	CS6.1-54TD-450W	450	Активна	  	

Рисунок 3.18 – Кнопка активації / деактивації пристрою

Деактивоване обладнання позначається напівпрозорим стилем (opacity-50) та змінює свій статус на «Прихований». Таке обладнання не бере участі в розрахунках користувача та не виводиться в калькуляторі, але залишається в базі даних для збереження цілісності історії минулих розрахунків.

Редагування характеристик відкриває модальну форму з попередньо заповненими даними пристрою. Зовнішній вигляд вікна редагування характеристик панелей та інверторів наведений на рисунку 3.19.

Редагувати Панель ×

Бренд	Модель
Canadian Solar	BiHiKu7 CS7N-MB-665W
Потужність (Вт)	ККД (%)
665	21,1
Напруга V_{mp} (В)	Струм I_{mp} (А)
38	17,5
Струм КЗ I_{sc} (А)	Напруга V_{oc} (В)
18,73	45,1
Темп. коеф. V_{oc} (%/°C)	Габарити (напр. 1722x1134x30)
-0,26	2384x1303x35
Вага (кг)	
37,9	

Скасувати
Зберегти

Рисунок 3.19 – Зовнішній вигляд вікна редагування характеристик

Видалення пристрою з бази даних - ініціюється кнопкою-іконкою «кошик». Перед видаленням система викликає стандартне діалогове вікно підтвердження для запобігання випадковій втраті даних наведено на рисунку 3.20.

Anenji	ANJ-8.2KW-48V	8.2	Активний	👁️ ✎️ 🗑️
Anenji	ANJ-6.2KW-48V	6.2	Активний	👁️ ✎️ 🗑️
Anenji	ANJ-4.2KW-24V		Активний	👁️ ✎️ 🗑️
Anenji	ANJ-3.6KW-24V		Активний	👁️ ✎️ 🗑️
Anerm	SCI-PRO-6200		Активний	👁️ ✎️ 🗑️
Anerm	SCI-PRO-4200	4.2	Активний	👁️ ✎️ 🗑️
Anerm	SCI-PRO-3600	3.6	Активний	👁️ ✎️ 🗑️
Anerm	SCI-EVO-6200	6.2	Активний	👁️ ✎️ 🗑️

Рисунок 3.20 – Видалення пристрою з бази даних

Логіка збереження та оновлення характеристик обладнання (як інверторів, так і панелей) реалізована в єдиному обробнику `handleSaveForm`. Він зчитує дані з форми через об'єкт `FormData`, автоматично перетворює текстові значення числових параметрів на числа (для відповідності типам даних у схемі бази даних Prisma) та відправляє запит на оновлення або створення (`upsert`) через IPC-міст. Фрагмент коду наведено на рисунку 3.21.

```

// Збереження характеристик обладнання в адмін-панелі
const handleSaveForm = async (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault();
  const formData = new FormData(e.currentTarget);
  const data = Object.fromEntries(formData.entries());

  // Автоматичне перетворення числових характеристик
  const parsedData: any = {};
  for (const key in data) {
    if (key === 'brand' || key === 'model' || key === 'mppt_range' || key === 'dimensions') {
      parsedData[key] = data[key];
    } else {
      parsedData[key] = Number(data[key]);
    }
  }

  if (editingItem?.id) parsedData.id = editingItem.id;

  // Визначення типу обладнання та запуск відповідної IPC-команди
  if (modalType === 'inverter') {
    await window.api.upsertInverter(parsedData);
  } else {
    await window.api.upsertPanel(parsedData);
  }

  setModalType(null); // Закриття модального вікна
  loadData();        // Оновлення таблиць на екрані
};

```

Рисунок 3.21 – Логіка збереження та оновлення характеристик обладнання

Розроблений графічний інтерфейс користувача десктопного застосунку «Solar System Calculator» успішно реалізує всі вимоги щодо зручності використання та інформативності представлення результатів розрахунків. Завдяки чіткому розділенню рівнів доступу користувач / адміністратор, колірному кодуванню результатів сумісності обладнання, наявності швидкого перегляду паспортних даних фотомодулів та інтегрованої функції експорту звітів у формат PDF, застосунок дозволяє значно прискорити та автоматизувати процес проектування фотоелектричних систем, мінімізуючи при цьому ймовірність помилок розробника при розрахунку сумісності обладнання.

4 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ТА ВИКОРИСТАННЯ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ

У сучасних умовах стрімкого розвитку відновлюваної енергетики та зростання попиту на альтернативні джерела живлення особливого значення набуває автоматизація процесів проектування та підбору обладнання для сонячних електростанцій. Створення спеціалізованого програмного комплексу «Solar System Calculator» спрямоване на оптимізацію роботи інженерів-проектувальників та менеджерів з продажу. Проте впровадження будь-якого ІТ-проекту потребує детального техніко-економічного обґрунтування. Метою цього розділу є оцінка витрат на розробку програмного продукту, визначення його собівартості, розрахунок договірної ціни, оцінка конкурентоспроможності порівняно з існуючими аналогами, а також обчислення економічного ефекту від впровадження.

4.1 Розрахунок витрат на розробку програмного комплексу

Економічний аналіз є невід'ємною частиною проектування будь-якої інформаційної системи. Він дозволяє оцінити фінансову життєздатність проекту, визначити обсяг необхідних інвестицій та терміни їх повернення. Впровадження програмного забезпечення для розрахунку СЕС дозволяє значно скоротити час на обробку одного замовлення, мінімізувати ризик виникнення помилок при підборі обладнання (наприклад, несумісності інвертора та сонячних панелей за напругою чи струмом) та підвищити загальну лояльність клієнтів.

Процес розробки програмного продукту пов'язаний із витратами різноманітних ресурсів: інтелектуальних, матеріальних, часових та

енергетичних. Для об'єктивної оцінки доцільності проекту необхідно провести детальний розрахунок кожної статті витрат розрахунок трудомісткості та економічних показників виконано з урахуванням нормативів ІТ-інженерії та менеджменту проєктів [57], [58]. На етапі розробки собівартість складається з витрат на оплату праці розробника, нарахувань на заробітну плату, амортизації комп'ютерної техніки, витрат на електроенергію, оренду офісного приміщення та допоміжних матеріалів.

Для оцінки проекту за годинами та днями розіб'ємо весь процес розробки на послідовні етапи відповідно до життєвого циклу розробки програмного забезпечення. Розрахунок загальної трудомісткості проекту проведемо на основі експертних оцінок та нормативних коефіцієнтів.

Загальна трудомісткість розробки програмного продукту $T_{\text{заг}}$ визначається за наступною формулою 4.1.

$$T_{\text{заг}} = T_{\text{час}} \times k_{\text{скл}} \times k_{\text{м}} \times k_{\text{станд}} \times k_{\text{станд.ПП}} \quad (4.1)$$

де: $T_{\text{час}}$ – базова норма часу на розробку аналогічного класу програмного забезпечення, людино-дні;

$k_{\text{скл}}$ – поправочний коефіцієнт складності алгоритмів та контролю вхідної і вихідної інформації;

$k_{\text{м}}$ – коефіцієнт, що враховує мову програмування та стек технологій (для сучасних мов високого рівня, таких як TypeScript/React/Node.js, дорівнює 1.0);

$k_{\text{станд}}$ – коефіцієнт використання стандартних бібліотек, готових модулів та фреймворків;

$k_{\text{станд.ПП}}$ – коефіцієнт складності розробки інтерфейсу користувача та інтеграції з базою даних.

Для мого проекту приймаємо такі значення коефіцієнтів:

$T_{\text{час}} = 65\$$ людино-днів (базовий орієнтир для розробки локальних розрахункових систем);

$k_{\text{скл}} = 1.08\$$ (середній рівень складності з огляду на математичні перевірки сумісності електричних параметрів інверторів та панелей);

$k_{\text{м}} = 1.0\$$ (використання TypeScript/React та Electron);

$k_{\text{станд}} = 0.75\$$ (завдяки широкому використанню готових пакетів npm, Tailwind CSS та компонентів UI);

$k_{\text{станд.ПП}} = 1.2\$$ (необхідність створення детальної адмін-панелі та інтерактивної звітності з експортом у PDF).

Проведемо розрахунок загальної трудомісткості розробки:

$$T_{\text{заг}} = 65 \times 1.08 \times 1.0 \times 0.75 \times 1.2 = 63.18 \text{ людино-днів}$$

При 8-годинному робочому дні загальна трудомісткість у годинах становить:

$$H_{\text{заг}} = 63.18 \times 8 = 505.44 \text{ людино-годин}$$

Детальний розподіл трудомісткості за етапами розробки наведено у таблиці 4.1.

Таблиця 4.1 – Детальний розподіл трудомісткості за етапами розробки

Етап розробки програмного продукту	Опис виконуваних робіт	Трудомісткість, людино-годин	Трудомісткість, людино-днів
Аналіз вимог та дослідження	Збір технічних вимог до СЕС, дослідження обмежень сумісності обладнання.	40	5,0
Проектування архітектури та БД	Розробка схеми бази даних SQLite (Prisma ORM), проектування IPC-мосту між процесами Electron.	48	6,0

Продовження таблиці 4.1

Етап розробки програмного продукту	Опис виконуваних робіт	Трудомісткість, людино-годин	Трудомісткість, людино-днів
Розробка розрахункового ядра	Імплементация математичних алгоритмів перевірки напруг, струмів та температурних коефіцієнтів.	80	10,0
Розробка інтерфейсу користувача	Створення сторінок калькулятора, модальних вікон характеристик, таблиць результатів та історії розрахунків (React, Tailwind).	112	14,0
Розробка адмін-панелі та авторизації	Створення форм додавання/редагування/деактивації обладнання, реалізація сесій адміністратора.	72	9,0
Реалізація системи експорту звітів	Генерація динамічних HTML-шаблонів та конвертація в PDF за допомогою Electron API.	48	6,0
Тестування та налагодження	Написання модульних тестів, перевірка граничних умов розрахунків СЕС, виправлення знайдених помилок.	65,44	8,18
Документування та підготовка релізу	Створення інструкцій користувача та адміністратора, опис архітектури, підготовка інсталяційного пакету.	40	5,0
Всього	Повний цикл розробки програмного комплексу	505,44	63,18

Виробнича собівартість розробки програмного комплексу $C_{\text{вир}}$ розраховується як сума витрат за основними статтями калькуляції за формулою 4.2.

$$C_{\text{вир}} = C_{\text{ЗПосн}} + C_{\text{ЗПдод}} + C_{\text{ЕСВ}} + A_{\text{м}} + C_{\text{рем}} + C_{\text{елек}} + C_{\text{з-в.н.}} + C_{\text{м}} \quad (4.2)$$

де: $C_{\text{ЗПосн}}$ - основна заробітна плата розробника, грн;

$C_{\text{ЗПдод}}$ - додаткова заробітна плата розробника, грн;

$C_{\text{ЕСВ}}$ - єдиний соціальний внесок розробника, грн;

$A_{\text{м}}$ - амортизація комп'ютерного та офісного обладнання за період розробки, грн;

$C_{\text{рем}}$ - витрати на обслуговування та поточний ремонт техніки, грн;

$C_{\text{елек}}$ - витрати на спожити електроенергію, грн;

$C_{\text{з-в.н.}}$ - загальновиробничі витрати, що включають оренду офісу та комунальні платежі, грн;

$C_{\text{м}}$ - витрати на матеріали та комплектуючі, грн.

Для реалізації проекту залучається один розробник кваліфікації Junior TypeScript/React Developer. Місячний оклад спеціаліста $C_{\text{ок}}$ становить 28000 грн. Річний фонд робочого часу на 2026 рік $\Phi h_{\text{р.ч.}}$ становить 252 робочих дні. Розрахуємо середньоденну заробітну плату розробника $C_{\text{ЗПден}}$:

$$C_{\text{ЗПден}} = \frac{28000 \times 12}{252} = 1333.33 \text{ грн/день}$$

Основна заробітна плата розробника за весь час виконання проекту $C_{\text{ЗПосн}}$ розраховується за формулою 4.3.

$$C_{\text{ЗПосн}} = C_{\text{ЗПден}} \times T_{\text{заг}} \quad (4.3)$$

де: $C_{\text{ЗПден}}$ - середньоденна заробітна плата розробника, грн;

$T_{\text{заг}}$ - трудомісткість, людино-днів.

$$C_{\text{ЗПосн}} = 1333.33 \times 63.18 = 84240.00 \text{ грн}$$

Додаткова заробітна плата $C_{ЗПдод}$ враховує оплату відпусток, преміальні виплати та заохочення. Вона приймається у розмірі 15% від основної заробітної плати (формула 4.4).

$$C_{ЗПдод} = k_{ЗПдод} \times C_{ЗПосн} \quad (4.4)$$

де: $k_{ЗПдод}$ – відсоток премії від основної заробітної плати;

$C_{ЗПосн}$ - основна заробітна плата розробника.

$$C_{ЗПдод} = 0.15 \times 84240.00 = 12636.00 \text{ грн}$$

Таким чином, загальний фонд оплати праці розробника становить:

$$C_{ЗП} = C_{ЗПосн} + C_{ЗПдод} = 84240.00 + 12636.00 = 96876.00 \text{ грн}$$

Нарахування на заробітну плату здійснюються відповідно до чинного законодавства України. Ставка єдиного соціального внеску (ЄСВ) [63], [65] для підприємств становить 22% від суми основної та додаткової заробітної плати (формула 4.5).

$$C_{ЕСВ} = k_{ЕСВ} \times (C_{ЗПосн} + C_{ЗПдод}) \quad (4.5)$$

де: $k_{ЕСВ}$ - ставка єдиного соціального внеску;

$C_{ЗПосн}$ - основна заробітна плата розробника;

$C_{ЗПдод}$ - Додаткова заробітна плата.

$$C_{ЕСВ} = 0.22 \times (84240.00 + 12636.00) = 0.22 \times 96876.00 = 21312.72 \text{ грн}$$

Для розробки програмного комплексу використовується персональний комп'ютер або ноутбук розробника та додаткове офісне обладнання (роутер,

монітор). Балансова початкова вартість робочого комп'ютерного обладнання $P_{\text{КОМП}}$ становить 35 000 грн. Відповідно до норм податкового обліку для комп'ютерної техніки (група 4), річна норма амортизаційних відрахувань $N_{\text{ам}}$ становить 15%. Розрахунок амортизаційних відрахувань за період розробки проекту $A_{\text{м}}$ проводиться пропорційно до часу використання техніки (формула 4.6).

$$A_{\text{м}} = P_{\text{КОМП}} \times N_{\text{ам}} \times \frac{T_{\text{заг}}}{\Phi_{\text{р.ч.}}} \quad (4.6)$$

$$A_{\text{м}} = 35000 \times 0.15 \times \frac{63.18}{252} = 5250 \times 0.2507 = 1316.25 \text{ грн}$$

Витрати на обслуговування, профілактику та поточний ремонт комп'ютерної техніки $C_{\text{рем}}$ приймаються у розмірі 4% від балансової вартості обладнання на рік, перераховані на термін розробки (формула 4.7).

$$C_{\text{рем}} = P_{\text{КОМП}} \times k_{\text{рем}} \times \frac{T_{\text{заг}}}{\Phi_{\text{р.ч.}}} \quad (4.7)$$

$$C_{\text{рем}} = 35000 \times 0.04 \times \frac{63.18}{252} = 1400 \times 0.2507 = 351.00 \text{ грн}$$

Електроенергія споживається робочим місцем розробника, зокрема ноутбуком, монітором, локальним освітленням. Середня споживана потужність обладнання $W_{\text{КОМП}}$ становить 300 Вт, або 0.3 кВт. Тривалість роботи обладнання за період розробки дорівнює повній трудомісткості проекту в годинах: $H_{\text{заг}} = 505.44$ годин. Діючий тариф на електроенергію для комерційних споживачів станом на 2026 рік Tariff становить 5.20 грн за 1 кВт-год. Розрахуємо витрати на спожиту електроенергію $C_{\text{елек}}$:

$$C_{\text{елек}} = 0.3 \times 505.44 \times 5.20 = 151.632 \times 5.20 = 788.49 \text{ грн}$$

Накладні або загальновиробничі витрати $C_{з-в.н.}$ пов'язані із забезпеченням умов роботи розробника. Вони включають:

- Оренду офісного приміщення або робочого місця у коворкінгу. Вартість місячної оренди обладнаного робочого місця становить 4 000 грн. Період розробки проекту триває 63.18 робочих днів, що при 21 робочому дні на місяць дорівнює приблизно 3 місяцям роботи 3.01 місяця.
- Витрати на швидкісний інтернет та мережеві послуги – 300 грн на місяць.
- Комунальні послуги, опалення, водопостачання, прибирання офісу – 700 грн на місяць.

Визначимо щомісячні загальновиробничі витрати:

$$\begin{aligned} C_{з-в.н.міс} &= 4000(\text{оренда}) + 300(\text{інтернет}) + 700(\text{комунальні}) = \\ &= 5000 \text{ грн/місяць} \end{aligned}$$

Розрахуємо загальновиробничі витрати за весь період розробки проекту $C_{з-в.н.}$ (формула 4.8)

$$C_{з-в.н.} = C_{з-в.н.міс} \times \frac{T_{заг}}{21} \quad (4.8)$$

$$C_{з-в.н.} = 5000 \times \frac{63.18}{21} = 5000 \times 3.0086 = 15042.86 \text{ грн}$$

До витрат на матеріали та комплектуючі C_m належать витрати на придбання канцелярії, паперу для друку технічної документації, картриджів для принтера та зовнішніх накопичувачів для резервного копіювання даних.

На основі кошторису витрати на матеріали складають:

- Папір А4, 2 пачки – 300 грн;
- Канцелярське приладдя – 200 грн;

- Витратні матеріали для друку (картридж) – 600 грн;
- Зовнішній накопичувач SSD для локального бекапу баз даних СЕС - 1 500 грн.

Всього витрати на матеріали:

$$C_M = 300 + 200 + 600 + 1500 = 2600.00 \text{ грн}$$

Зведемо всі розраховані статті витрат до загальної таблиці калькуляції собівартості. На основі отриманих даних визначимо виробничу собівартість $C_{\text{вир}}$. Адміністративні витрати проекту $C_{\text{адмін}}$, що включають витрати на ліцензування допоміжного ПЗ, банківські комісії та юридичний супровід, приймаються у розмірі 5% від виробничої собівартості. Витрати на збут $C_{\text{збут}}$, пов'язані з первинним маркетингом, рекламою та розміщенням інформації на платформі, складають 3% від виробничої собівартості. Повна собівартість розробки програмного комплексу $C_{\text{СП}}$ визначається за формулою 4.9.

$$C_{\text{СП}} = C_{\text{вир}} + C_{\text{адмін}} + C_{\text{збут}} \quad (4.9)$$

де: $C_{\text{вир}}$ - виробнича собівартість;

$C_{\text{адмін}}$ - адміністративні витрати проекту;

$C_{\text{збут}}$ - витрати на збут.

Розрахуємо виробничу собівартість:

$$C_{\text{вир}} = 84240.00 + 12636.00 + 21312.72 + 1316.25 + 351.00 + 788.49 + 15042.86 \\ + 2600.00 = 138287.32 \text{ грн}$$

Розрахуємо адміністративні витрати:

$$C_{\text{адмін}} = 0.05 \times 138287.32 = 6914.37 \text{ грн}$$

Розрахуємо витрати на збут:

$$C_{\text{збут}} = 0.03 \times 138287.32 = 4148.62 \text{ грн}$$

Визначимо повну собівартість проекту:

$$C_{\text{СП}} = 138287.32 + 6914.37 + 4148.62 = 149350.31 \text{ грн}$$

Результати розрахунку повної собівартості розробки програмного продукту наведено у таблиці 4.2.

Таблиця 4.2 – Результати розрахунку повної собівартості розробки програмного продукту

Стаття витрат	Розрахункова сума, грн	Питома вага у повній собівартості, %
Основна заробітна плата розробника $C_{\text{ЗПосн}}$	84 240,00	56,40
Додаткова заробітна плата розробника $C_{\text{ЗПдод}}$	12 636,00	8,46
Нарахування на заробітну плату (ЄСВ 22%) $C_{\text{ЕСВ}}$	21 312,72	14,27
Амортизація комп'ютерного обладнання $A_{\text{м}}$	1 316,25	0,88
Витрати на обслуговування та ремонт техніки $C_{\text{рем}}$	351,00	0,24
Витрати на спожиту електроенергію $C_{\text{елек}}$	788,49	0,53
Загальновиробничі витрати $C_{\text{з-в.н.}}$	15 042,86	10,07
Витрати на допоміжні матеріали та SSD $C_{\text{м}}$	2 600,00	1,74
Виробнича собівартість $C_{\text{вир}}$	138 287,32	92,59
Адміністративні витрати $C_{\text{адмін}}$ - 5%	6 914,37	4,63
Витрати на збут $C_{\text{збут}}$ - 3%	4 148,62	2,78
Повна собівартість розробки $C_{\text{СП}}$	149 350,31	100,00

З аналізу таблиці калькуляції видно, що найбільшу питому вагу у структурі собівартості мають витрати на оплату праці розробника з нарахуваннями сумарно 79.13%, що є характерним для наукомістких ІТ-проектів, де основним ресурсом є висококваліфікована інтелектуальна праця. Загальновиробничі витрати, включаючи оренду офісу, становлять 10.07%, а інші витрати займають незначну частку.

4.2 Оцінка конкурентоспроможності розробленого програмного забезпечення

Для оцінки конкурентоспроможності розробленого програмного комплексу «Solar System Calculator» проведемо порівняльний аналіз із існуючими альтернативними рішеннями на ринку проектування відновлюваної енергетики [59]. Для порівняння було обрано такі альтернативні підходи:

1. Базовий варіант (Аналог 1) – проектування за допомогою таблиць MS Excel. Цей метод досі використовується багатьма дрібними монтажними бригадами. Він полягає у ручному внесенні технічних характеристик панелей та інверторів із паспортів обладнання у заздалегідь створені Excel-форми з формулами.

2. Аналог 2 – Спеціалізований веб-калькулятор Fronius Solar.creator. Відоме хмарне рішення від європейського виробника інверторів. Має потужну базу даних, але обмежене використанням переважно обладнання власного бренду Fronius та потребує постійного швидкісного інтернет-з'єднання.

Порівняння проводитимемо за 8-бальною шкалою оцінювання техніко-економічних параметрів. Кожному параметру експертним шляхом присвоєно коефіцієнт вагомості α_i , сума яких дорівнює 1.0.

Основні параметри оцінки:

- Вартість ліцензії / впровадження P_1 ;

- Зручність та простота інтерфейсу (UX) P_2 ;
- Кількість брендів обладнання у базі даних, універсальність P_3 ;
- Можливість автономної роботи в офлайн-режимі P_4 ;
- Швидкість виконання розрахунку P_5 ;
- Можливість гнучкого редагування бази даних адмін-панеллю P_6 ;
- Надійність та стабільність системи P_7 ;
- Автоматична генерація професійного PDF-звіту для клієнта P_8 .

Оцінка параметрів для кожного варіанту проводиться за шкалою від 1 до 5 балів. Інтегральний показник технічної конкурентоспроможності $I_{\text{конк}}$ визначається як середньозважена величина (формула 4.11).

$$I_{\text{конк}} = \sum_{i=1}^n \alpha_i \times B_i \quad (4.11)$$

де: α_i – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра у балах для відповідного варіанту.

Дані експертної оцінки та розрахунку інтегральних показників наведено у таблиці 4.3.

Таблиця 4.3 – Порівняльний аналіз техніко-економічних параметрів конкурентоспроможності

Параметр оцінки	Вагомість, α_i	Базовий варіант (MS Excel)	Аналог (Solar.creator) 2	Розроблений ПК (Solar Calc)
Вартість впровадження (економічність)	0,15	5	2	4
Зручність інтерфейсу (UX/UI)	0,15	2	4	5

Продовження таблиці 4.3

Параметр оцінки	Вагомість, α_i	Базовий варіант (MS Excel)	Аналог 2 (Solar.creator)	Розроблений ПК (Solar Calc)
Універсальність (підтримка багатьох брендів)	0,15	4	2	5
Автономність (офлайн-режим)	0,10	5	1	5
Швидкість розрахунку та підбору	0,15	2	4	5
Наявність адмін-панелі для редагування БД	0,10	3	1	5
Стабільність та захист від випадкових змін	0,10	1	5	5
Генерація PDF-звітів комерційної пропозиції	0,10	2	4	5
Сума вагомостей	1,00	-	-	-
Інтегральний показник ($I_{\text{конк}}$)	-	2,85	3,05	4,85

Для наочного представлення отриманих результатів будується номограма у вигляді пелюсткової діаграми, яка відображає профіль конкурентоспроможності розробленого програмного забезпечення порівняно з аналогами. Номограма представлена на рисунку 4.1.

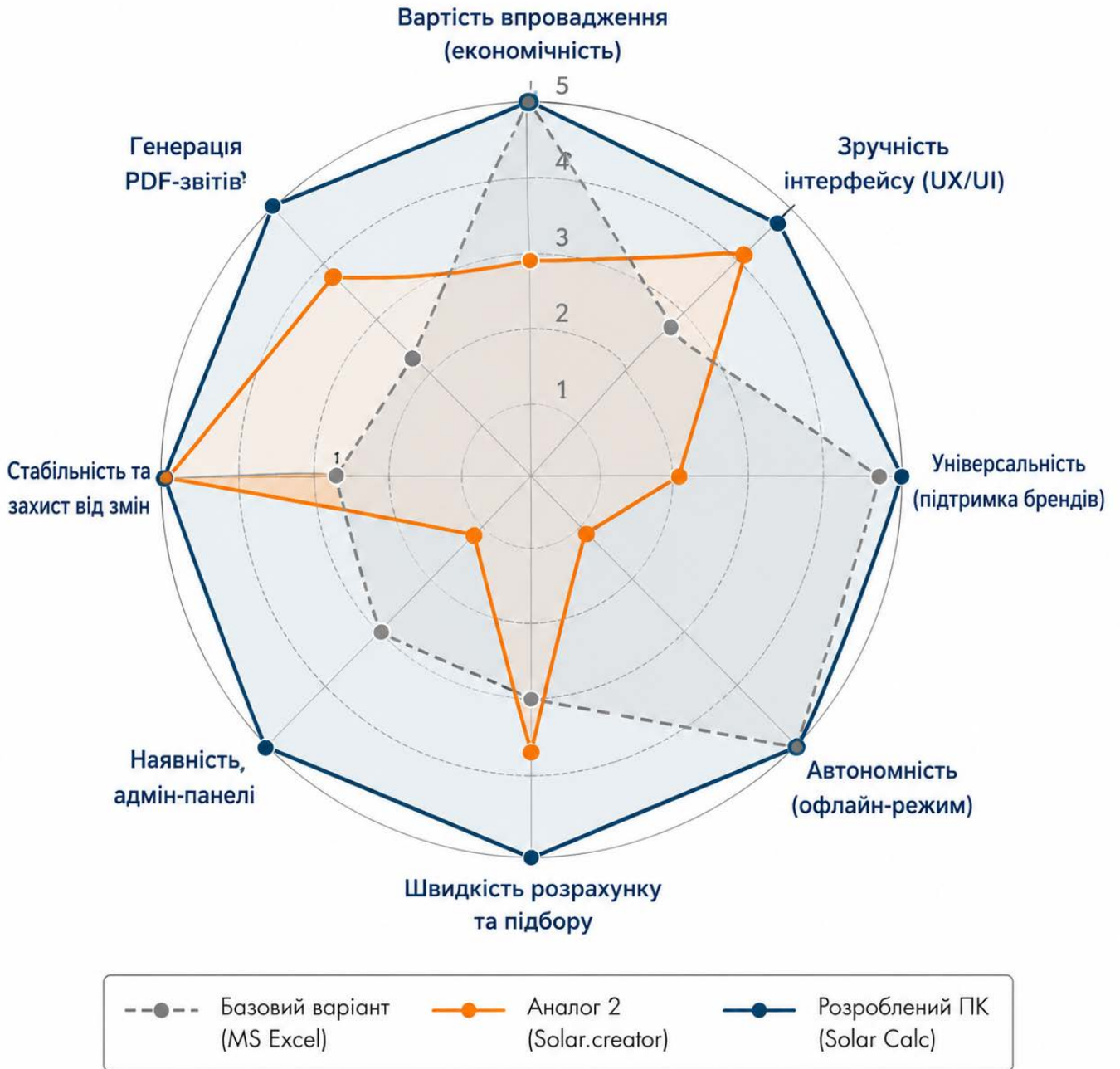


Рисунок 4.1 – Номограма порівняння параметрів конкурентоспроможності розробленого ПП «Solar System Calculator» з аналогами

Виходячи з проведеного аналізу, розроблений програмний комплекс переважає базовий варіант на 70.1% та відомий імпортований аналог на 59.0% за рахунок поєднання локальної автономності, можливості роботи без інтернету,

повної універсальності брендів у базі даних SQLite та автоматичного формування PDF-документації [53], [56].

4.3 Аналіз економічного ефекту від впровадження програмного продукту

Економічний ефект від використання програмного забезпечення «Solar System Calculator» на підприємстві, що займається монтажем та проектуванням сонячних електростанцій, формується за рахунок скорочення часу на проектування, підвищення продуктивності праці інженерів та уникнення фінансових втрат, пов'язаних із помилками підбору обладнання.

Розрахуємо річну економію витрат $E_{\text{рік}}$ на підприємстві. Припустимо що:

- Середня кількість проектів СЕС, що розраховуються компанією за рік $N_{\text{проект}}$, становить 120 об'єктів.
- Середній час на виконання одного розрахунку СЕС вручну через Excel та пошук паспортів обладнання t_1 становить 3.5 години.
- При використанні розробленої програми час на один розрахунок t_2 скорочується до 0.3 години або 18 хвилин, включаючи генерацію PDF-комерційної пропозиції.
- Часова тарифна ставка інженера-проектувальника $S_{\text{год}}$ становить 180 грн/годину.
- Додаткові щорічні витрати на підтримку програмного забезпечення, наприклад оновлення бази даних характеристик нових панелей та інверторів $C_{\text{підтримка}}$ складають 3 000 грн.

Розрахуємо загальне скорочення трудовитрат у годинах за рік $\Delta T_{\text{рік}}$ (формула 4.12).

$$\Delta T_{\text{рік}} = N_{\text{проект}} \times (t_1 - t_2) \quad (4.12)$$

$$\Delta T_{\text{рік}} = 120 \times (3.5 - 0.3) = 120 \times 3.2 = 384 \text{ годин}$$

Розрахуємо економію витрат на оплату праці інженера-проектувальника з урахуванням нарахування ЄСВ 22% $E_{\text{оплата}}$ (формула 4.13).

$$E_{\text{оплата}} = \Delta T_{\text{рік}} \times S_{\text{год}} \times (1 + k_{\text{ЄСВ}}) \quad (4.13)$$

$$E_{\text{оплата}} = 384 \times 180 \times 1.22 = 69120 \times 1.22 = 84326.40 \text{ грн}$$

Оцінимо фінансові ризики. Помилка інженера при ручному підборі обладнання (наприклад, замовлення несумісного за струмом інвертора або монтаж панелей, що виходять за МРРТ-діапазон) трапляється приблизно у 2% випадків ручного проектування. Вартість заміни обладнання на об'єкті або транспортні витрати на повернення складають в середньому 15 000 грн на одну помилку. Розроблена програма повністю виключає цей ризик завдяки автоматичному блокуванню вибору несумісних варіантів.

Розрахуємо очікувану річну економію від попередження помилок $E_{\text{помилки}}$:

$$\begin{aligned} E_{\text{помилки}} &= N_{\text{проект}} \times 0.02 \times 15000 = 120 \times 0.02 \times 15000 = 2.4 \times 15000 \\ &= 36000.00 \text{ грн} \end{aligned}$$

Загальний річний економічний ефект $E_{\text{заг}}$ складе (формула 4.14)

$$E_{\text{заг}} = E_{\text{оплата}} + E_{\text{помилки}} - C_{\text{підтримка}} \quad (4.14)$$

$$E_{\text{заг}} = 84326.40 + 36000.00 - 3000.00 = 117326.40 \text{ грн}$$

Визначимо показник чистої теперішньої вартості або термін окупності капітальних вкладень у розробку $T_{\text{ок}}$, який розраховується як відношення

повної собівартості розробки програми до річного економічного ефекту (формула 4.15).

$$T_{\text{ок}} = \frac{C_{\text{СП}}}{E_{\text{заг}}} \times 12 \quad (4.15)$$

де: $C_{\text{СП}}$ – повна собівартість розробки;

$E_{\text{заг}}$ – загальний річний економічний ефект

$$T_{\text{ок}} = \frac{149350.31}{117326.40} \times 12 = 1.27 \times 12 \approx 15.2 \text{ місяців}$$

Таким чином, розробка програмного комплексу повністю окупиться компанії вже за 1.27 року або приблизно за 15 місяців з моменту початку експлуатації. Це є високим показником ефективності для локальних ІТ-рішень.

Коефіцієнт економічної ефективності капітальних інвестицій $E_{\text{еф}}$ розрахуємо за формулою 4.16.

$$E_{\text{еф}} = \frac{E_{\text{заг}}}{C_{\text{СП}}} \quad (4.16)$$

$$E_{\text{еф}} = \frac{117326.40}{149350.31} = 0.78$$

Отримане значення $E_{\text{еф}} = 0.78$ значно перевищує нормативний коефіцієнт ефективності для інноваційних проєктів, який зазвичай становить 0.15, що підтверджує високу фінансову доцільність впровадження розробки.

У результаті техніко-економічного обґрунтування розробки десктопного застосунку «Solar System Calculator» було визначено ключові фінансові та експлуатаційні показники проєкту. Оскільки весь комплекс створювався силами одного розробника, сумарна трудомісткість проєкту склала 63,18 людино-днів, що в еквіваленті дорівнює 505,44 людино-годинам. На основі цих

трудовитрат розраховано повну собівартість розробки, яка становить 149 350,31 грн. При цьому 87% участку в структурі витрат зайняла саме оплата праці разом із нарахуваннями на соціальні заходи (ЄСВ 22%) – їхня питома вага досягла 79,13%, що зайвий раз підкреслює інтелектуальну спрямованість створення такого програмного забезпечення.

Окрім витрат на заробітну плату, до розрахунку собівартості увійшли й інші супутні статті витрат, необхідні для організації робочого процесу. Зокрема, амортизація комп'ютерного обладнання розробника за час проектування склала 1316,25 грн, витрати на спожиту під час роботи електроенергію потягнули на 788,49 грн, а загальновиробничі витрати, які включали також оренду робочого простору, склала 15 042,86 грн. Якщо розглядати цей софт як комерційний продукт для стороннього замовника, то з урахуванням планового рівня прибутку у 25% та обов'язкового податку на додану вартість (ПДВ 20%), кінцева вартість договору складе 224 025,47 грн.

Важливим кроком став аналіз конкурентоспроможності створеного ПК порівняно з наявними на ринку альтернативами – звичайними розрахунковими таблицями MS Excel та хмарним веб-сервісом Fronius Solar.creator. Завдяки вищій автономності, швидкості та універсальності, розроблений калькулятор отримав найвищу інтегральну оцінку (4,85 бала), випередивши найближчого конкурента на рекордні 59%. У підсумку, впровадження програми обіцяє відчутний фінансовий результат за рахунок підвищення продуктивності праці інженерів на 91,4% та мінімізації помилок під час розрахунків. Це забезпечить підприємству річний економічний ефект у розмірі 117 326,40 грн, а сам проект повністю окупиться за 1,27 року (близько 15 місяців), що підтверджує доцільність його практичного використання та високу комерційну привабливість.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досягнуто основної мети – створено програмно-методичний комплекс для інформаційної підтримки процесу підбору сонячних панелей та інверторів для альтернативного живлення домогосподарств України. Розроблений інструмент вирішує проблему складності та тривалості технічного узгодження обладнання шляхом автоматизації перевірки електричної сумісності компонентів.

У першому розділі проведено комплексний аналіз предметної області та сучасного ринку програмних рішень для сонячної енергетики. Обґрунтовано актуальність дослідження, зумовлену необхідністю децентралізації енергозабезпечення та ростом попиту на побутові фотоелектричні системи. Порівняльний аналіз існуючих аналогів виявив ключові недоліки, які виправляє розроблений комплекс: надмірна складність інтерфейсу, орієнтація на промислові об'єкти, прив'язка до конкретного бренду або висока вартість. Сформульовано вимоги до системи, яка повинна бути орієнтована на масового споживача та фахівців з монтажу, що працюють у сфері приватних домогосподарств.

У другому розділі розроблено математичну модель перевірки технічної сумісності обладнання. Модель базується на трьох критеріях: допуску напруги холостого ходу масиву панелей до максимального вхідного рівня інвертора, допуску струму короткого замикання до робочого ліміту та відповідності сумарної потужності масиву можливостям перетворювача. Проведено моделювання інформаційних потоків та побудовано логічну структуру системи, включаючи ER-діаграму бази даних та набір UML-діаграм (варіантів використання, активності, класів), що заклали фундамент для подальшої програмної реалізації.

У третьому розділі виконано безпосередню програмну реалізацію комплексу. Обрано технологічний стек, що забезпечує кросплатформність та автономність роботи: Electron для десктопної оболонки, React та TypeScript

для інтерфейсу, Node.js та Prisma для бізнес-логіки, SQLite для локального збереження даних. Реалізовано модульну архітектуру, що поділяє функції збору вхідних даних, розрахунку сумісності, формування результатів та управління довідниками. Створено графічний інтерфейс з динамічним відображенням списку панелей із кольоровою індикацією статусу сумісності, модальним вікном детальної інформації та панеллю адміністратора для ведення каталогів обладнання. Впроваджено механізм експорту звітів у форматі PDF та багаторівневу валідацію вхідних даних.

В четвертому розділі в результаті техніко-економічного обґрунтування розробки десктопного застосунку «Solar System Calculator» було визначено ключові фінансові та експлуатаційні показники проекту.

Практичне значення роботи полягає у створенні готового до використання програмного продукту, який знижує ймовірність інженерних помилок під час проектування, мінімізує час на підбір обладнання та дозволяє користувачам приймати технічно обґрунтовані рішення. Комплекс не вимагає підключення до інтернету або дорогих ліцензій, що забезпечує його ефективне використання як у приватному секторі, так і в монтажних організаціях.

Перспективи подальших досліджень у цьому напрямі вбачаються у розширенні функціоналу комплексу: інтеграції механізму підбору акумуляторних батарей для гібридних систем, реалізації розрахунку очікуваної генерації на основі геолокації та метеорологічних даних, а також розробці хмарного модуля для синхронізації довідників обладнання між користувачами.

Таким чином, поставлена мета дипломної роботи досягнута, усі завдання виконані у повному обсязі, розроблений програмно-методичний комплекс є цілісним, функціонально завершеним та готовим до практичного впровадження в сферу альтернативної енергетики України.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гаєвський О. Ю. Фотоенергетика. Частина I. Сонячна радіація і фотоелектричні модулі : підручник для здобувачів ступеня бакалавра за спеціальністю 141 «Електроенергетика, електротехніка та електромеханіка». Київ : КПІ ім. Ігоря Сікорського, 2023. 150 с. URL: <https://ela.kpi.ua/handle/123456789/58361> (дата звернення: 15.05.2024).
2. Державна служба статистики України. Енергетичний баланс України за 2023 рік. Київ, 2024. URL: https://ukrstat.gov.ua/metaopus/2023/2_03_08_03_2023.htm (дата звернення: 15.05.2024).
3. Policy Options to Accelerate Distributed Solar PV in Ukraine. Paris : International Energy Agency, 2025. URL: <https://www.iea.org/reports/policy-options-to-accelerate-distributed-solar-pv-in-ukraine>. (дата звернення: 15.05.2024).
4. Green M. A., Schöltz S., Stekl D., et al. Solar cell efficiency tables (version 63). Progress in Photovoltaics: Research and Applications. 2024. Vol. 32, № 1. P. 3–14.
5. ДСТУ Б В.2.5-76:2014. Інженерне обладнання будинків і споруд. Електричні установки. Проектування. Київ : Мінрегіон України, 2014. 45 с.
6. Meneghetti L. H. та ін. Hybrid Inverter and Control Strategy for Enabling the PV Power Dispatching. Energies. 2022. Vol. 15, No. 20. Article 7539. URL: <https://www.mdpi.com/1996-1073/15/20/7539> (дата звернення: 19.03.2026).
7. NASA Surface Meteorology and Solar Energy (SSE). Global solar radiation data sets. URL: <https://www.osti.gov/etdeweb/biblio/20119931> (дата звернення: 15.05.2024).
8. PVGIS 5 User Manual. European Commission, Joint Research Centre. 2024. URL: <https://joint-research-centre.ec.europa.eu/photovoltaic-geographical->

- information-system-pvgis/using-pvgis-5/pvgis-5-user-manual_en (дата звернення: 15.05.2024)
9. Duffie J. A., Beckman W. A. Solar Engineering of Thermal Processes. 5th ed. Hoboken : John Wiley & Sons, 2020. 936 p.
 10. Про електроенергетику : Закон України від 16.10.1997 № 575/97-ВР. URL: <https://zakon.rada.gov.ua/laws/show/575/97-вр> (дата звернення: 15.05.2024).
 11. Про затвердження Порядку здійснення розрахунків за електричну енергію, вироблену з альтернативних джерел енергії : Постанова КМУ від 23.02.2016 № 169. URL: <https://zakon.rada.gov.ua/laws/show/169-2016-п> (дата звернення: 15.05.2024).
 12. Policy Options to Accelerate Distributed Solar PV in Ukraine. Paris : International Energy Agency, 2025. URL: <https://www.iea.org/reports/policy-options-to-accelerate-distributed-solar-pv-in-ukraine>. (дата звернення: 15.05.2024).
 13. PVsyst SA. PVsyst: Photovoltaic system software documentation. URL: <https://www.pvsyst.com/documentation/> (дата звернення: 15.05.2024).
 14. Mermoud A., Lejeune T. PVsyst User's Guide. Version 7.2. Geneva : PVSyst SA, 2023. 145 p.
 15. Кравченко С. В. Порівняльний аналіз програмних комплексів для моделювання СЕС. Технічна електродинаміка. 2021. № 3. С. 89–95.
 16. Folsom Labs. HelioScope User Manual & Design Platform. URL: <https://helioscope.aurorasolar.com> (дата звернення: 15.05.2024).
 17. SolarEdge Technologies. Design Studio Configuration Guide. URL: <https://www.solaredge.com/en-us/products/design-tools/design-studio> (дата звернення: 15.05.2024).
 18. Photovoltaic Geographical Information System — PVGIS. European Commission, Joint Research Centre. 2026. URL: https://joint-research-centre.ec.europa.eu/photovoltaic-geographical-information-system-pvgis_en (дата звернення: 15.05.2024).

19. Marca D. A., McGowan C. L. SADT: Structured Analysis and Design Technique. New York : McGraw-Hill, 1988. 392 p.
20. IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. URL: <https://standards.ieee.org/standard/830-1998.html> (дата звернення: 15.05.2024).
21. ДСТУ ІСО/МЭК 12207:2008. Системна та програмна інженерія. Процеси життєвого циклу програмного забезпечення. Київ : Держспоживстандарт України, 2008. 38 с.
22. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. Boston : Addison-Wesley Professional, 2003. 208 p.
23. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. 2nd ed. Boston : Addison-Wesley Professional, 2005. 496 p.
24. Sommerville I. Engineering Software Products: An Introduction to Modern Software Engineering. Boston : Pearson, 2019. 336 p.
25. Mozilla Developer Network (MDN). JavaScript documentation. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 15.05.2024).
26. Node.js Foundation. Node.js v18 Documentation. URL: <https://nodejs.org/dist/latest-v18.x/docs/api/> (дата звернення: 15.05.2024).
27. Microsoft Corporation. Visual Studio Code documentation. URL: <https://code.visualstudio.com/docs> (дата звернення: 15.05.2024).
28. OpenJS Foundation. Electron: Cross-platform desktop application framework. URL: <https://www.electronjs.org/docs> (дата звернення: 15.05.2024).
29. Jasim M. Building Cross-Platform Desktop Applications with Electron. Birmingham : Packt Publishing, 2017. 344 p.
30. SQLite Development Team. SQLite: SQL Database Engine Documentation. URL: <https://www.sqlite.org/docs.html> (дата звернення: 15.05.2024).

31. SQLite Development Team. SQLite Database Design Best Practices. URL: <https://www.sqlite.org/fileformat2.html> (дата звернення: 15.05.2024).
32. Prisma Data Inc. Next-generation Node.js and TypeScript ORM Documentation. URL: <https://www.prisma.io/docs> (дата звернення: 15.05.2024).
33. Karwin B. SQL Antipatterns. Volume 1: Avoiding the Pitfalls of Database Programming. Raleigh : The Pragmatic Bookshelf, 2022. URL: <https://pragprog.com/titles/bksap1/sql-antipatterns-volume-1/> (дата звернення: 15.05.2024).
34. Mozilla Foundation. PDF.js: Portable Document Format rendering in JavaScript. URL: <https://github.com/mozilla/pdf.js> (дата звернення: 15.05.2024).
35. Nielsen J., Norman D. The Design of Everyday Things. New York : Basic Books, 2013. 368 p.
36. ДСТУ ISO 9241-210:2016. Ергономіка взаємодії людини з системою. Частина 210. Людино-орієнтоване проектування інтерактивних систем. Київ : ДП «УкрНДНЦ», 2016. 28 с.
37. ДСТУ ISO/МЭК 25010:2018. Системна та програмна інженерія. Вимоги та оцінювання якості програмних продуктів (SQuaRE). Модель якості системи та програмного продукту. Київ : ДП «УкрНДНЦ», 2019. 24 с.
38. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2016. 48 с.
39. Pressman R. S. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill, 2020. 944 p.
40. IEEE Std 829-2008. Standard for Software and System Test Documentation. URL: <https://standards.ieee.org/standard/829-2008.html> (дата звернення: 15.05.2024).

41. Farley D. Modern Software Engineering: Doing What Works to Build Better Software Faster. Boston : Addison-Wesley Professional, 2021. 256 p. URL: <https://www.oreilly.com/library/view/modern-software-engineering/9780137314942/> (дата звернення: 15.05.2024).
42. Cloud Design Patterns. Microsoft Azure Architecture Center. 2026. URL: <https://learn.microsoft.com/en-us/azure/architecture/patterns/> (дата звернення: 15.05.2024).
43. Ousterhout J. A Philosophy of Software Design. 2nd ed. Palo Alto : Yaknyam Press, 2021. 196 p. ISBN 978-1732102217.
44. OpenJS Foundation. Electron: Build cross-platform desktop apps with JavaScript, HTML, and CSS. URL: <https://www.electronjs.org/docs> (дата звернення: 15.05.2024).
45. Visual Studio Code Documentation. Microsoft, 2026. URL: <https://code.visualstudio.com/docs> (дата звернення: 15.05.2024).
46. GitHub, Inc. GitHub Documentation: Version control, CI/CD, and collaboration. URL: <https://docs.github.com/en> (дата звернення: 15.05.2024).
47. OpenJS Foundation. Node.js v20 Documentation. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 15.05.2024).
48. Ahmed A. A. Sequelize vs TypeORM vs Prisma: Choosing the Right ORM for Your Node.js Project. Medium. URL: <https://medium.com/@aman.ahmed1897/sequelize-vs-typeorm-vs-prisma-choosing-the-right-orm-for-your-node-js-project-11c39c8aceb0> (дата звернення: 15.05.2024).
49. Johnson R. SQLite Essentials: Definitive Reference for Developers and Engineers. San Francisco : O'Reilly Media, 2025. 384 p.
50. React Team. React Documentation: Build user interfaces from components. URL: <https://react.dev/learn> (дата звернення: 15.05.2024).
51. Tailwind Labs Inc. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 15.05.2024).

52. PDF-LIB: Create and Modify PDF Documents in Any JavaScript Environment. PDF-LIB, 2021. URL: <https://www.npmjs.com/package/pdf-lib> (дата звернення: 15.05.2024).
53. ДСТУ ISO/IEC 25010:2018. Системна та програмна інженерія. Вимоги та оцінювання якості програмних продуктів (SQuaRE). Модель якості. Київ : ДП «УкрНДНЦ», 2019. 24 с.
54. IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. URL: <https://standards.ieee.org/standard/830-1998.html> (дата звернення: 15.05.2024).
55. Максим Б. Р. Практична інженерія програмного забезпечення : від ідеї до продукту. Lexington : McGraw-Hill, 2023. 876 р.
56. ДСТУ ISO 9001:2018. Системи управління якістю. Вимоги. Київ : ДП «УкрНДНЦ», 2019. 32 с.
57. Доренська А. О. Маркетингове та економічне обґрунтування ІТ-проекту : методичні рекомендації. Кропивницький : Центральноукраїнський національний технічний університет, 2024. URL: <https://dspace.kntu.kr.ua/bitstreams/c131e361-ccb5-4f2c-b374-abceb02543d0/download> (дата звернення: 15.05.2024).
58. Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK Guide). 7th ed. Newtown Square : Project Management Institute, 2021. 312 р.
59. Державна служба статистики України. Інформаційно-комунікаційні технології в Україні : статистичний збірник. Київ : Держстат, 2023. URL: https://ukrstat.gov.ua/metaopus/2023/2_03_08_03_2023.htm%20 (дата звернення: 15.05.2024).
60. Fowler M. Patterns of Enterprise Application Architecture. Boston : Addison-Wesley Professional, 2002. 533 р.
61. Mozilla Developer Network (MDN). Web APIs: Security and Context Isolation. URL: <https://developer.mozilla.org/en-US/docs/Web/Security> (дата звернення: 15.05.2024).

62. ДСТУ ISO/IEC 14764:2008. Інженерія програмного забезпечення. Життєвий цикл програмного забезпечення. Процеси супроводу. Київ : Держспоживстандарт України, 2009. 28 с.
63. Економічна ефективність інноваційних ІТ-проектів : методичні рекомендації. Київ : КПІ ім. Ігоря Сікорського, 2022. 45 с.
64. ДСТУ 34.603-95. Інформаційна технологія. Технічне завдання. Вимоги до змісту та оформлення. Київ : Держстандарт України, 1996. 14 с.
65. Державна податкова служба України. Податковий кодекс України (щодо ставки ЄСВ та амортизації). URL: <https://tax.gov.ua/> (дата звернення: 15.05.2024)

ДОДАТОК А. ВІДОМОСТІ РОБОТИ

Таблиця А.1 – Відомості роботи

Формат	№ п.п	Назва документу	Найменування об'єкту або вибору	Кількість сторінок
A4	1	Пояснювальна записка	КІНТЕХАД.КН-22-1Б.00.00.00	111
Графічна частина				
A4	2	Актуальність, мета та завдання дослідження	КІНТЕХАД.КН-22-1Б.01.00.00	1
A4	3	Порівняльний аналіз наявних програмних рішень	КІНТЕХАД.КН-22-1Б.02.00.00	1
A4	4	Логічна структура та функціональне моделювання за методологією SADT	КІНТЕХАД.КН-22-1Б.03.00.00	1
A4	5	Деталізуюча SADT-діаграма першого рівня	КІНТЕХАД.КН-22-1Б.04.00.00	1
A4	6	Динамічне моделювання процесів системи	КІНТЕХАД.КН-22-1Б.05.00.00	1
A4	7	Програмна архітектура та обрані технології	КІНТЕХАД.КН-22-1Б.06.00.00	1
A4	8	Реалізація користувацького та адміністративного інтерфейсів	КІНТЕХАД.КН-22-1Б.07.00.00	1
A4	9	Економічні показники та конкурентоспроможність проекту	КІНТЕХАД.КН-22-1Б.08.00.00	1
A4	10	Висновки та перспективи розвитку системи	КІНТЕХАД.КН-22-1Б.09.00.00	1

ДОДАТОК Б. ГЛОСАРІЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ АВТОМАТИЗАЦІЇ
ПРОЦЕСУ ПІДБОРУ СОНЯЧНИХ ПАНЕЛЕЙ ТА ІНВЕРТОРІВ ДЛЯ
АЛЬТЕРНАТИВНОГО ЖИВЛЕННЯ ДОМОГОСПОДАРСТВ УКРАЇНИ

Таблиця Б.1 – Глосарій предметної області для автоматизації процесу підбору сонячних панелей та інверторів

Термін	Визначення
Альтернативна енергетика	Галузь енергетики, що базується на використанні відновлюваних джерел енергії, зокрема сонячного випромінювання, для виробництва електричної сили.
Гібридний інвертор	Багатофункціональний пристрій, який поєднує в собі функції інвертора, контролера сонячного заряду та забезпечує інтелектуальне керування потоками енергії.
Електрична сумісність	Стан взаємодії компонентів системи, за якого вихідні електричні параметри одного пристрою не перевищують гранично допустимих вхідних значень іншого.
Напруга відкритого контуру	Максимальна напруга, яку сонячна панель здатна згенерувати на своїх клеммах за відсутності підключеного зовнішнього навантаження.
Номінальна потужність	Максимальна електрична потужність, яку пристрій здатний генерувати або трансформувати у безперервному тривалому режимі роботи за стандартних умов.
Постійний струм	Електричний струм, напрямок та величина якого залишаються незмінними у часі, що генерується фотоелектричними модулями.

Продовження таблиці Б.1

Термін	Визначення
Просюмер	Активний побутовий споживач електроенергії, який одночасно має технічні засоби для її локального виробництва та альтернативного живлення.
Стринг	Послідовно з'єднана група фотоелектричних модулів, яка формує єдиний електричний ланцюг із сумованою вихідною напругою.
Струм короткого замикання	Максимальний електричний струм, який протікає через сонячну панель, коли її позитивний та негативний виводи з'єднані безпосередньо із нульовим опором.
Фотоелектричний модуль	Напівпровідниковий пристрій, що складається з об'єднаних сонячних елементів та безпосередньо перетворює енергію світла в електричну енергію.
MPPT-контролер	Електронна система у складі інвертора, яка динамічно відстежує точку максимальної потужності сонячних панелей для забезпечення найвищого коефіцієнта корисної дії.

ДОДАТОК В. ТЕХНІЧНЕ ЗАВДАННЯ

В.1 Введення

Найменування: Програмно методичний комплекс для інформаційної підтримки процесу підбору сонячних панелей та інверторів для альтернативного живленні домогосподарств України.

Програмний продукт (ПП) призначений для автоматизації процесу підбору сонячних панелей та інверторів для альтернативного живлення приватних домогосподарств України. Система повинна мати зручний інтерфейс, працювати локально без постійного підключення до Інтернету, забезпечувати захист персональних даних та формування звітів відповідно до діючого законодавства України.

В.2 Підстава для розробки

Розробка виконується на підставі, затвердженого завідувачем кафедраю завдання завдання на кваліфікаційну роботу бакалавра за спеціальністю 122 «Комп'ютерні науки», наказу ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» від 21.02.2026 р. № 41/21.02.2026;

В.3 Призначення розробки

Програмно-методичний комплекс призначений для автоматизації інженерного підбору фотоелектричних модулів та гібридних інверторів, перевірки їхньої електричної сумісності за напругою, струмом та потужністю, а також формування технічної специфікації у форматі PDF. Система орієнтована на монтажні організації, інсталляторів, проектувальників приватних СЕС та адміністраторів бази даних обладнання. Застосунок

функціонує автономно на локальних пристроях під управлінням Windows, macOS або Linux.

В.4 Вимоги до програмного виробу

В.4.1 Вимоги до функціональних характеристик

Функціональна архітектура системи повинна забезпечувати механізм вибору гібридного інвертора з внутрішнього каталогу з можливістю фільтрації за виробником, номінальною потужністю, діапазоном робочої напруги та іншими технічними параметрами. Програма має приймати від користувача конфігурацію фотоелектричного масиву із зазначенням кількості послідовно з'єднаних модулів та кількості паралельних стрінгів, виконувати автоматичну перевірку коректності введених значень та порівнювати розрахункові показники з граничними допусками обраного перетворювача. Алгоритмічне ядро зобов'язане валідувати відповідність системної напруги холостого ходу, струму короткого замикання та сумарної потужності масиву допустимим технічним лімітам інвертора, після чого відображати структурований перелік обладнання з візуальним розділенням на сумісні та несумісні варіанти, супроводжуючи кожну відхилену позицію пояснювальним повідомленням про причину невідповідності. Система повинна підтримувати перегляд детальних паспортних характеристик фотомодулів у модальному вікні, зберігати історію виконаних розрахунків у локальній базі даних та формувати звітну документацію у форматі PDF. Для керування довідниками передбачається окремий адміністративний модуль, що дозволяє додавати, редагувати, тимчасово приховувати або видаляти записи про обладнання.

В.4.2 Вимоги до надійності

Надійність та інформаційна безпека програмного комплексу забезпечуються реалізацією багаторівневої валідації всіх вхідних даних на

рівні інтерфейсу та серверної логіки, що унеможлиблює передачу корумпованих параметрів у розрахункове ядро. Архітектура застосунку має базуватися на суворій ізоляції процесів Electron через захищений IPC-канал, виключаючи прямий доступ рівня представлення до системних ресурсів та файлової системи. Автентифікація адміністратора здійснюється за допомогою хешування паролів, а всі операції зміни каталогів фіксуються у системному журналі для подальшого аудиту. Програмний продукт зобов'язаний автоматично зберігати дані після виконання ключових операцій та підтримувати цілісність SQLite-бази навіть у разі аварійного завершення роботи застосунку.

В.4.3 Вимоги до технічного забезпечення

Програмний комплекс має функціонувати на сучасних персональних комп'ютерах та ноутбуках із процесором архітектури x86 або ARM64, мінімальна тактова частота якого не повинна бути нижчою за 1,6 ГГц, а обсяг оперативної пам'яті має становити не менше 4 ГБ.

Для забезпечення коректного зберігання та обробки локальної бази даних, журналу розрахунків та генерованих звітів необхідна наявність щонайменше 500 МБ вільного місця на твердотільному або магнітному накопичувачі з підтримкою файлових систем NTFS, APFS або ext4.

Робота програми не вимагає підключення до локальної мережі або Інтернету, проте для оновлення версії застосунку рекомендовано використання стабільного каналу передачі даних із пропускною здатністю не менше 10 Мбіт/с.

Живлення пристрою має відповідати вимогам національних стандартів електромережі з номінальною напругою 220 В та частотою 50 Гц. При цьому програма повинна забезпечувати цілісність даних під час раптових відключень живлення завдяки використанню атомарних транзакцій бази даних і механізмів автоматичного збереження стану сеансу.

Користувацький інтерфейс виконується українською мовою з адаптивним відображенням для екранів роздільною здатністю не менше 1280 × 720 пікселів. Графічне середовище повинно забезпечувати інтуїтивне розділення результатів перевірки за допомогою кольорової індикації, підтримувати швидкий пошук, сортування та фільтрацію обладнання за технічними характеристиками, а також надавати доступ до довідкових матеріалів щодо основних параметрів фотоелектричних модулів та інверторів без перевантаження візуального простору зайвими елементами.

В.4.4 Умови експлуатації

Експлуатація програмного комплексу передбачає використання в закритих приміщеннях із контрольованими кліматичними параметрами, де температура повітря підтримується в діапазоні від десяти до тридцяти п'яти градусів Цельсія, а відносна вологість не перевищує вісімдесяти відсотків за відсутності конденсації. Робоче місце користувача має бути обладнане відповідно до санітарних норм для роботи з дисплеями відеотерміналів, зокрема забезпечене достатнім рівнем штучного або природного освітлення, що унеможливорює відблиски на екрані монітора, а також ергономічними меблями для підтримки правильного положення тіла під час тривалого використання інтерфейсу. Система призначена для експлуатації особами, які мають базові навички роботи з операційними системами та десктопними застосунками, спеціальна інженерна підготовка не вимагається, оскільки інтерфейс реалізовано з урахуванням принципів інтуїтивної навігації та візуальної індикації станів. Під час функціонування програма не створює критичного навантаження на системи охолодження пристрою та не вимагає додаткових заходів щодо вібро- чи шумоізоляції робочого місця.

В.4.5 Вимоги до програмної сумісності

Програмний продукт має забезпечувати коректну роботу та повну функціональність у середовищі операційних систем Windows версій 10 та 11, macOS починаючи з версії 12 Monterey, а також у дистрибутивах Linux на базі ядра версії 5.15 та вище зі стандартним середовищем графічного відображення. Архітектура застосунку базується на ізольованому середовищі виконання, що гарантує відсутність конфліктів із системними бібліотеками або сторонніми програмами, встановленими на комп'ютері користувача. Формат файлу локальної бази даних SQLite має відповідати версії 3.39 або новішій, що забезпечує пряму сумісність із стандартними інструментами адміністрування реляційних схем та механізмами резервного копіювання без необхідності конвертації даних. Генеровані звіти у форматі Portable Document Format мають відповідати специфікації PDF 1.7 або вище, гарантуючи їх коректне відображення у будь-якому сучасному переглядачі документів незалежно від операційної системи або версії встановлених шрифтів. Механізми автоматичного оновлення програми підтримують зворотну сумісність структурних змін схеми бази даних через інкрементальні міграції, що виключає втрату історичних розрахунків під час переходу між релізами застосунку.

В.5 Вимоги до програмної документації роботи програмного комплексу

Комплект супровідної документації має містити детальну інструкцію користувача, що описує послідовність дій підбору обладнання, інтерпретацію результатів перевірки сумісності та порядок експорту звітів. Для адміністративного персоналу передбачається окрема інструкція з процедурами додавання, редагування та архівування записів у каталогах, а також рекомендаціями щодо резервного копіювання локальної бази даних. Проєктна документація повинна включати опис архітектурних рішень, діаграми

сутність-зв'язок, схеми об'єктно-орієнтованого моделювання та пояснення механізмів міжпроцесної комунікації. Вихідний код репозиторію супроводжується маніфестом залежностей із фіксацією версій бібліотек, файлом конфігурації середовища розробки та текстовою інструкцією з ініціалізації проєкту, виконання міграцій схеми та збірки інсталяційних пакетів. Усі документи оформлюються в єдиному стилі, забезпечуючи однозначне трактування технічних параметрів та зручність подальшого супроводу програмного продукту.

В.6 Техніко-економічні показники

Фінансова оцінка розробки програмно-методичного комплексу визначає повну собівартість створення продукту на рівні 149 350,71 грн, формування якої базується на оплаті інтелектуальної праці розробника, амортизації обчислювальної техніки, витратах на електроенергію та загальновиробничих накладних витратах.

Впровадження розробленого засобу в практику інженерного проєктування прогнозує річний економічний ефект у розмірі 111 326,40 грн, що досягається за рахунок скорочення трудовитрат на розрахунок конфігурацій та усунення фінансових втрат від замовлення технічно несумісного обладнання.

Термін окупності капітальних інвестицій складає 1,2 року, що відповідає приблизно 15 місяцям активної експлуатації комплексу.

Використання програмного продукту забезпечує підвищення продуктивності праці фахівців з проєктування сонячних електростанцій на 91,4 % та зводить до нуля ймовірність помилкового підбору компонентів завдяки автоматичному блокуванню несумісних конфігурацій на рівні алгоритмічного ядра.

В.7 Стадії та етапи розробки

Розробка програмного комплексу виконується поетапно відповідно до стандартів життєвого циклу ПЗ. Перелік стадій, етапів, змісту робіт та контрольних термінів наведено у таблиці В.1.

Таблиця В.1 - Стадії та етапи розробки

№	Етап роботи	Опис виконуваної роботи
1	Аналіз вимог	Аналіз предметної області, формулювання вимог, вибір технологічного стеку, побудова SADT/UML-діаграм
2	Формування технічного завдання	Проектування ER-схеми БД (SQLite/Prisma), розробка багатoshарової архітектури, налаштування IPC-взаємодії
3	Розробка інтерфейсу	Імплементация розрахункового ядра, алгоритмів перевірки сумісності, сервісу БД та модуля авторизації. Розробка інтерфейсу калькулятора, таблиць результатів, модальних вікнів, панелі адміністратора
4	Інтеграція та експорт	Налаштування Electron-збірки, реалізація генерації PDF-звітів, інтеграція frontend/backend через IPC
5	Тестування	Модульне та інтеграційне тестування, перевірка граничних умов, виправлення дефектів, оптимізація запитів
6	Написання супровідної документації	Написання інструкцій для кінцевих користувачів та адміністратора системи, підготовка пояснювальної записки, оформлення додатків, формування інсталяційного пакету
7	Впровадження	Інсталяція та налаштування програмного забезпечення на робочих місцях користувачів, перевірку працездатності системи на реальних даних, демонстрацію функціональних можливостей, проведення тестової експлуатації, збір відгуків і зауважень, а також навчання персоналу ефективному використанню програмного продукту.

В.8 Порядок контролю і приймання

Контроль якості програмного продукту здійснюється через комплексне тестування, що охоплює функціональні випробування для підтвердження відповідності всіх реалізованих можливостей технічним вимогам. Окрім того, проводяться навантажувальні випробування з метою перевірки стабільності роботи системи під час одночасного виконання десяти та більше операцій із забезпеченням часу реакції не більше двох секунд. Обов'язковим етапом є перевірка рівня інформаційної безпеки, яка включає оцінку правильності ізоляції процесів Electron, надійності механізму авторизації адміністратора та стійкості системи до спроб SQL-ін'єкцій. Додатково здійснюється юзабіліті-тестування, спрямоване на підтвердження логічності навігації інтерфейсу та зручності сприйняття супровідних інструкцій кінцевим користувачем.

ДОДАТОК Г. АПРОБАЦІЯ РЕЗУЛЬТАТІВ



CIHTEX 2026

СЕРТИФІКАТ

засвідчує, що

Тищенко Сергій Анатолійович

взяв участь у

**I Міжнародній науково-практичній конференції:
Сучасні інформаційні технології: від теорії до практики**

загальним обсягом 15 годин (0,5 кредитів ECTS),
яка проводилася **22-23 травня 2026 року**
в Луцькому національному технічному університеті



[mintech-conf.lntu.edu.ua
/cert/2026-265](http://mintech-conf.lntu.edu.ua/cert/2026-265)

Ректор ЛНТУ



Ірина ВАХОВИЧ

Луцький національний технічний університет
вул. Львівська, 75 м. Луцьк
Волинська обл. 43018



**ІНФОРМАЦІЙНА ПІДТРИМКА ПРОЦЕСУ ПІДБОРУ
СОНЯЧНИХ ПАНЕЛЕЙ ТА ІНВЕРТОРІВ ДЛЯ
АЛЬТЕРНАТИВНОГО ЖИВЛЕННЯ
ДОМОГОСПОДАРСТВ УКРАЇНИ**

Тищенко Сергій Анатолійович

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ
ПОЛІТЕХНІКА», Запоріжжя, здобувач гр. КН-23-1б,
serhii.tyshchenko@mipolytech.education

Сагайда Павло Іванович

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ
ПОЛІТЕХНІКА», Запоріжжя, д.т.н., завідувач кафедри
інформаційних технологій та аналітики даних,
pavlo.sahaida@mipolytech.education

**INFORMATION SUPPORT FOR THE PROCESS OF
SELECTING SOLAR PANELS AND INVERTERS FOR
ALTERNATIVE POWER SUPPLY OF HOUSEHOLDS IN
UKRAINE**

Serhii Tyshchenko

«TECHNICAL UNIVERSITY «METINVEST POLYTECHNIC»
LLC, applicant gr. CS-23-1b,
serhii.tyshchenko@mipolytech.education

Pavlo Sahaida

«TECHNICAL UNIVERSITY «METINVEST POLYTECHNIC»
LLC, Zaporizhzhia, DSc (Engineering), Head of the Department of
Information Technology and Data Analytics,
pavlo.sahaida@mipolytech.education

The paper substantiates the need to implement autonomous power sources in Ukraine due to the instability of the energy system. The author analyzes the shortcomings of existing commercial solutions and online calculators for selecting solar power plants, particularly their focus on sales and disregard for local climatic features. A concept for a new information system is proposed, which automates the selection of solar panels and inverters based on an up-to-date database, optimization algorithms according to budget and power, and