МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ

ім. Ю.М. Потебні

ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

перший (бакалаврський)
(рівень вищої освіти)

на тему **Програмне забезпечення для розрахунку потреби у запасних частинах**

Виконав: студент 4 курсу, групи 6.1211-пзс
спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)
освітньої програми Програмне забезпечення систем
(код і назва освітньої програми)
Л. А. Власова
(ініціали та прізвище)
Керівник к.т.н., доцент О. М. Міхайлуца
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)
Рецензент Директор ТОВ «Дискус»
Р.О. Лютий
(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя
2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ

ім. Ю.М. Потебні

ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра електроніки, інформаційних систем та програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Спеціальність 121 Інженерія програмного забезпечення
(код та назва)
Освітня програма Програмне забезпечення систем
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри Т.В. Критська
“ 01 ” березня 2025 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Власовій Лілії Андріївні
(прізвище, ім’я, по батькові)

1. Тема роботи Програмне забезпечення для розрахунку потреби у запасних частинах

керівник роботи Міхайлуца Олена Миколаївна, к.т.н., доцент
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ЗНУ від 27.12.2024 №2029-с

2. Строк подання студентом кваліфікаційної роботи
3. Вихідні дані бакалаврської роботи
   - комплект нормативних документів ;
   - технічне завдання до роботи.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   - огляд та збір літератури стосовно теми кваліфікаційної роботи;
   - огляд та аналіз існуючих рішень та аналогів;
   - дослідження проблеми розрахунку потреби у запасних чатинах;
   - створення програмного продукту та його опис;
   - дослідження поставленої проблеми та розробка висновків.
5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
   слайдів презентації

6. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.03.2025

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів бакалаврської роботи	Примітка
1	Аналіз предметної області	02.03.25	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	05.03.25	виконано
3	Аналіз існуючих методів рішення	10.03.25	виконано
4	Дослідження засобів реалізації програмного забезпечення	15.03.25	виконано
5	Дослідження засобів реалізації сайту для розрахунку потреби у запасних частинах.	25.03.25	виконано
6	Узгодження подальших дій з науковим керівником	26.03.25	виконано
7	Проектування архітектури додатку для розрахунку потреби у запасних частинах.	27.03-01.04.25	виконано
8	Програмна реалізація серверної частини додатку для розрахунку потреби у запасних частинах.	02.04-25.04.25	виконано
9	Реалізація клієнтської частини додатку для розрахунку потреби у запасних частинах.	25.04-13.05.25	виконано
10	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	14.05.25	виконано
11	Реалізація користувацького інтерфейсу додатку для розрахунку потреби у запасних частинах	15.05-30.05.25	виконано
12	Перевірка роботоздатності проектів	02.06-04.06.25	виконано
13	Оформлення звіту	05.06-12.06.25	виконано
14	Оформлення презентації. Отримання рецензій від опонентів.	13.06-15.06.25	виконано

Студент _____ Власова Л. А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Міхайлуца О. М.
(підпис) (прізвище та ініціали)

Нормоконтроль пройдено

Нормоконтролер _____ Скрипник І.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок — 67

Рисунків — 21

Таблиць — 4

Джерел — 21

Додатків — 2

Власова Л. А. Програмне забезпечення для розрахунку потреби у запасних частинах: кваліфікаційна робота бакалавра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник О. М. Міхайлуца. Запоріжжя: ЗНУ, 2025. 67 с.

Мета роботи полягає у вивченні стратегій та методів розрахунку необхідної кількості запасних частин, їх оптимізації за допомогою розробки програмного забезпечення, що містить в собі весь необхідний функціонал.

В процесі роботи було досліджено, спроектовано та створено програмне забезпечення для автоматизації управління запасами на машинобудівних підприємствах Запорізького регіону. Це дослідження є вкрай актуальним з огляду на складність виробничих процесів у машинобудуванні та високу конкуренцію на світовому ринку. Запропонована система дозволить замінити ручні розрахунки та забезпечить точне визначення необхідної кількості запасних частин та матеріалів.

Апробація одержаних результатів. Результати роботи були опубліковані на Всеукраїнській науково-практичній конференції «Актуальні питання сталого науково-технічного та соціально-економічного розвитку регіонів України», XIV Всеукраїнській конференції «Молоді вчені 2024 — від теорії до практики», I Міжнародній науково-практичній конференції Інженерного навчально наукового інституту ім. Ю. М. Потебні Запорізького національного університету «Інженерні інновації та розбудова національної економіки». Акт впровадження в освітній процес додається.

Ключові слова: *Django, інформаційна система, метод домінуючих послідовностей, оптимізація запасів.*

ABSTRACT

Pages —67

Figures —21

Tables —4

References —21

Appendices —2

Vlasova L. A. Software for Calculating Spare Parts Requirements: bachelor's thesis in specialty 121 "Software Engineering" / Science manager O. M. Mikhailutsa. Zaporizhzhia: ZNU, 2025. p.

The goal of the work is to study the strategies and methods for calculating the required number of spare parts, their optimization through the development of software that contains all the necessary functionalities.

During the work process, software was researched, designed, and created for automating inventory management in machine engineering enterprises in the Zaporizhzhia region. This research is extremely relevant given the complexity of production processes in machine engineering and the high competition in the global market. The proposed system will replace manual calculations and ensure accurate determination of the required number of spare parts and materials.

Validation of Results The results of this work were published at the All-Ukrainian Scientific and Practical Conference "Topical Issues of Sustainable Scientific, Technical, and Socio-Economic Development of Ukrainian Regions," the XIV All-Ukrainian Conference "Young Scientists 2024 — From Theory to Practice," and the I International Scientific and Practical Conference of the Engineering Educational and Scientific Institute named after Yu. M. Potebnia of Zaporizhzhia National University "Engineering Innovations and the Development of the National Economy." An act of implementation into the educational process is attached.

Keywords: *Django, information system, dominant sequences method, inventory optimization.*

ЗМІСТ

ВСТУП 7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 13
1.1 Огляд літературних джерел 13
1.2 Аналіз існуючих рішень 15
1.2.1 Мануальний розрахунок 15
1.2.2 Альтернативні способи розрахунку 19
1.2.3 Спеціалізовані програмні системи 22
1.3 Постановка завдання 28
1.4 Висновки до розділу 1 28
2 РЕАЛІЗАЦІЯ ПРОГРАМНОГО РІШЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ЗАПАСІВ 30
2.1 Вимоги до програмного продукту 30
2.2 Проєктування системи 31
2.3 Архітектура додатку 36
2.4 Середовище розробки PyCharm 42
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ 45
3.1 Проектування додатку 45
3.2 Розробка програмного забезпечення 46
3.3 Тестування програмного продукту 56
ВИСНОВКИ 64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 65
ДОДАТОК А КРЕСЛЕНИК МОТАЛКИ ГАРЯЧОЇ ШТАБИ ДЛЯ ЗМОТУВАННЯ РУЛОНІВ З ВІСЬМОМА ФОРМУЮЧИМИ РОЛИКАМИ. 68
ДОДАТОК Б КРЕСЛЕНИК РЕДУКТОРА ПРИВОДА МОТАЛКИ ГАРЯЧОЇ СМУГИ ДЛЯ ЗМОТУВАННЯ РУЛОНІВ З ВІСЬМОМА ФОРМУЮЧИМИ РОЛИКАМИ 69

ВСТУП

Актуальність дослідження

Жодна держава не зможе існувати без підприємств, які підтримують і піднімають економіку країни. Від них багато в чому залежить об’єм ВВП, роль держави на міжнародному ринку, робочі місця та рівень життя населення.

Кожне підприємство має обладнання, елементи якого під час експлуатації можуть втрачати свої початкові характеристики. Для безперервного функціонування підприємств періодично проводиться капітальний ремонт обладнання від якого залежить забезпечення відповідної надійності машин. Тому потрібно правильно розрахувати потребу у запасних частинах, що в свою чергу залежить від потрібної надійності обладнання. Для того щоб провести капітальний ремонт — потрібно закупити запасні частини, а для закупки відповідно правильно визначити потрібну їх кількість.

Якщо змінних елементів буде недостатньо, і це буде виявлено на стадії проведення ремонтних робіт, то підприємство втратить кошти. У такій ситуації є два варіанти розвитку подій. Перший — докупити запасні частини, яких не вистачило. Але на замовлення, закупку і доставку(що теж має свою вартість) відповідних деталей потрібен додатковий час. Відповідно обладнання не працюватиме, а підприємство буде потерпати від збитків у даний період. Другий варіант — залишити все як є , і просто не замінювати елементи яких не вистачило, але це теж має негативні наслідки. У цьому випадку надійність відповідного обладнання знижується, що збільшить ймовірність поломок у міжремонтний період, або збільшить вартість наступного капітального ремонту. Відсоток зниження надійності залежить від конкретного обладнання, деталей яких не вистачило та грамотного розподілення наявних елементів, враховуючи їх зношуваність у різних вузлах конструкції. Оскільки несправність одного елемента може призвести до ланцюгової реакції пошкоджень інших компонентів, загальна працездатність конструкції опиняється під загрозою. Відсутність якісного капітального

ремонту ускладнює прогнозування таких пошкоджень, що, в свою чергу, вимагає термінової заміни деталей. Цей сценарій не лише призводить до простоїв у роботі, але й створює додаткові витрати на незаплановані ремонтні роботи. Тож підприємство понесе збитки за ремонт обладнання та період його проведення, або за додаткові запасні частини під час проведення наступного капітального ремонту.

Надмірне накопичення запасних частин може призвести до значних фінансових втрат для підприємства. Оборотний капітал, який міг би бути інвестований в розвиток виробництва або інші перспективні проекти, буде «заморожений» на складах. Крім того, навіть якщо деталі зберігаються для подальшого використання, існує ризик втрати їхніх властивостей з часом. Оскільки проведення дефектоскопії для оцінки стану кожної деталі може бути досить дорогим, підприємство ризикує витратити значні кошти на зберігання та перевірку запасних частин, які можуть виявитися непридатними для використання.

В умовах війни підприємства, особливо розташовані в прифронтових регіонах (таких як Запоріжжя, Дніпро, Кам'янське, Кривий Ріг, Харків, Полтава, Одеса, Суми), стикаються з високим ризиком пошкодження або повного знищення виробничих потужностей, у тому числі складів із запасними частинами. Склади з запасними частинами є вразливими об'єктами, і їхнє пошкодження може призвести до значних фінансових втрат. Тому, для мінімізації ризиків, підприємствам варто розглянути можливість оптимізації запасів, відмовившись від зберігання великої кількості запасних частин.

Щоб уникнути надмірних витрат на закупівлю та зберігання запасних частин, а також мінімізувати ризики, пов'язані з їх втратою внаслідок військових дій, необхідно впровадити ефективну систему управління запасами. Першим кроком є детальний аналіз експлуатаційної історії обладнання на основі агрегатного журналу, що дозволить визначити оптимальний обсяг запасних частин для проведення планових ремонтів.

У динамічному бізнес-середовищі, де конкуренція висока, а ситуація часто змінюється, ефективне управління запасами є одним із найважливіших факторів, що впливають на успіх компанії. Розробка спеціалізованого програмного забезпечення для оптимізації управління запасами змінних частин є вкрай актуальним завданням, оскільки:

- Оптимізація запасів дозволяє знизити витрати на зберігання, закупівлю та утилізацію непотрібних запасів.
- Завдяки своєчасному заповненню запасів можна забезпечити безперебійний виробничий процес та швидку доставку продукції клієнтам.
- Ефективне управління запасами сприяє підвищенню обігу оборотного капіталу та збільшенню прибутковості підприємства.
- Оптимізація запасів дозволяє знизити ризики, пов’язані з дефіцитом або надлишком запасів.
- Застосування програмного забезпечення дозволяє автоматизувати рутинні операції, підвищити точність обліку та аналізу даних.

Мета роботи

Мета роботи полягає у вивченні стратегій та методів розрахунку необхідної кількості запасних частин, їх оптимізації за допомогою розробки програмного забезпечення, що містить в собі весь необхідний функціонал.

Об’єкт дослідження

Об'єктом дослідження є процес розробки програмного забезпечення для розрахунку потреби у запасних частинах.

Предмет дослідження

Процес розрахунку потреби у запасних частинах на машинобудівному підприємстві з метою визначення оптимальних рівнів запасів та мінімізації пов'язаних з ними витрат.

Завдання

Провести детальний огляд сучасних підходів до управління запасами, методів прогнозування попиту та оптимізації рівнів запасів, вивчити існуючі програмні продукти для управління запасами та визначити їхні переваги та недоліки, розробити програмний продукт для автоматизації процесу управління запасами на виробничому підприємстві, який дозволить оптимізувати рівні запасів, мінімізувати витрати на зберігання та забезпечити безперебійність виробничого процесу.

Методи дослідження: загальнонаукові (емпіричні, комплексні й теоретичні) та спеціальні (графічні та метод компонентного аналізу).

Новизна дослідження: дослідження пропонує новий підхід до оптимізації управління запасами шляхом інтеграції класичного методу домінуючих послідовностей з індивідуальними характеристиками обладнання машинобудівної галузі.

Практичне значення одержаних результатів дослідження: Запропонована методика оптимізації запасів дозволяє підприємствам машинобудівної галузі точніше прогнозувати попит на запасні частини, зменшити ризик дефіциту та знизити витрати на зберігання, що в кінцевому рахунку призводить до підвищення конкурентоспроможності підприємств.

Також, дане програмне забезпечення може бути використане в навчальному процесі при опануванні курсів, в яких розглядається надійність машин і обладнання та методи і технології проведення ремонтів, в саме : «Монтаж, експлуатація та ремонт металургійного обладнання» та «Надійність металургічних систем».

Апробація результатів дослідження: Тези доповідей на конференціях:

- XIV Всеукраїнській конференції «Молоді вчені 2024 - від теорії до практики» за темою «Стратегії та методи розрахунку потреби в запасних частинах» [15].

- Міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи»(МН-2024), за темою «Застосування патернів проектування для розробки програмного комплексу з визначення

потреби у запасних частинах» [17].

- І Міжнародній науково-практичній конференції Інженерного навчально

наукового інституту ім. Ю. М. Потебні Запорізького національного університету «Інженерні інновації та розбудова національної економіки», за темою «Автоматизація розрахунків запасних частин для капітального ремонту: опис, можливості та перспективи розвитку» [18].

- XVIII Університетській науково-практичній конференції студентів, аспірантів, докторантів і молодих вчених «Молода наука-2025», за темою «Програмне забезпечення для розрахунку потреби у запасних частинах» [16].

Глосарій

Домінуючий вектор — кількість запасних частин кожного типу, де кожен елемент є окремим типом складових системи

Ймовірність відмови елемента — ймовірність того, що елемент відмовить у заданий час.

Капітальний ремонт — ремонт, що виконується для відновлення справності та повного або близького до повного ресурсу виробу з заміною або відновленням будь-яких його частин, включаючи базові. Показники значень щодо повного ресурсу, встановлюють у нормативно-технічній документації. ДСТУ 9050:2020.

Машинобудування — найважливіша комплексна галузь обробної промисловості, яка включає проєктування, виробництво й експлуатацію машин та інструментів.

Надійність — властивість технічних об'єктів зберігати протягом встановленого часу значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання та транспортування.

Оптимізація запасів — надходження матеріальних ресурсів оптимальними партіями, котрі забезпечують кількість елементів, яка за періодичного надходження забезпечує мінімальні сукупні витрати на організацію купівлі і зберігання запасів.

Потреба у запасних частинах — необхідна кількість запасних частин для проведення ремонтних робіт

Приріщення достатності — зміна ймовірності безвідмовної роботи, при зміні кількості запасних частин.

Ремонтний цикл — це сукупність різних видів планового ремонту, які повторюються і виконуються в передбаченій послідовності через встановлену однакову кількість годин роботи обладнання, що називаються міжремонтними періодами. Ремонтний цикл завершується капітальним ремонтом.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд літературних джерел

У час активного науково технічного розвитку, оптимізація процесів є однією з пріоритетних задач. А під час роботи будь яких виробництв використовується обладнання котре потрібно ремонтувати. Тому питання визначення потреби у запасних частинах завжди буде актуальним. У процесі підготовки дипломної роботи важливим завданням є аналіз літературних джерел, що дозволить глибше зрозуміти тему та визначити наявні підходи до розрахунку потреби у запасних частинах. Основна увага приділяється збору і аналізу літературних джерел, які становлять основу даної роботи. Метод розрахунку потреби у запасних частинах для капітального ремонту, котрий розробив Седуш В. Я., є основним у роботі, так як він ґрунтується на досягненні потрібної надійності обладнання. Також існує метод оцінки попиту на запасні частини на основі моделі суміші ймовірнісних розподілів, описаний у статті про закономірності та особливості формування потреби у запасних частинах на підприємствах автосервісу, але за його допомогою визначається економічна доцільність закупки елементів на склад, що не вирішує поставленої задачі [13].

Відкриті літературні джерела піддані детальному аналізу, зокрема зосередженням на темах надійності обладнання, ремонтів, визначення потреби у запасних частинах та способах їх визначення. Різні підходи до визначення потреби у запасних частинах, розглядаються зокрема з урахуванням їхнього впливу на ефективність вирішення задачі розрахунку.

Серед основних тем, виявлених під час аналізу літературних джерел, варто зазначити наступні. По-перше, опис методики розрахунку потреби у запасних частинах Седуша В.Я.. По-друге, дослідження різних існуючих способів проведення розрахунків даним методом, для забезпечення

оптимізації процесу розрахунку відносно ручного. По-третє, вивчення архітектури та підходів до розробки аналогічних систем.

Важливість дослідження та розробки програмного забезпечення для розрахунку потреби у запасних частинах, обумовлюється тим, що він стане допоміжним інструментом для проведення капітальних ремонтів обладнання підприємств різних типів. Так як основний акцент робиться на надійності обладнання. Вивчення різних методик щодо розробки систем ведення розрахунків ймовірностей, дозволяє обрати оптимальну архітектуру, вид та засоби реалізації програмного продукту.

Створення системи для розрахунку потреби у запасних частинах для проведення капітального ремонту може бути складною задачею, оскільки потребує розуміння не лише програмування, а й надійності обладнання та способу проведення розрахунку. Знання та розуміння його етапів і помилок що можуть виникати під час його проведення.

Також можуть виникати проблеми з правильною організацією запису, зберігання та роботи з даними. Використання спеціалізованого програмного продукту для забезпечення правильного виконання розрахунків, та ідентифікації машин, вузлів та елементів, враховуючи їх креслення, специфікації, моделі, тощо значно полегшить цю задачу. Методологія включає використання баз даних, онлайн-бібліотек та фреймворків. Під час відбору літературних джерел основними критеріями є актуальність, науковий рівень та авторитетність авторів.

Отже, розробка програмного забезпечення для розрахунку потреби в запасних частинах для проведення капітальних ремонтів, є актуальною оскільки дозволяє оптимізувати роботу підприємства, а саме:

- точніше прогнозувати попит на запасні частини,
- зменшити ризик дефіциту змінних елементів
- знизити витрати на зберігання змінних елементів, та знизити ризик їх знищення в наслідок воєнних дій
- підвищення конкурентоспроможності підприємств.

1.2 Аналіз існуючих рішень

За стандартними методами розрахунків визначають ймовірність відмови кожного елемента, який підлягає заміні під час проведення капітального ремонту. Після цього можна приступати до розрахунків.

Забезпеченість запасними частинами під час виконання капітального ремонту- запорука виконання робіт в заплановані терміни без додаткових простоїв. Прогнозована кількість запасних частин скорочує витрати на надлишковий запас, та виключає недостатню їх кількість. В рамках проведеного дослідження розглянуто основні способи прогнозування необхідної кількості запасних частин [3-7].

1.2.1 Мануальний розрахунок

Перший спосіб — це мануальний розрахунок. Нехай система містить *N* різних типів замінних елементів (підшипники, вали, шестерні та ін.), Кожен *i*-тий тип (*i* = 1,2, ..., *N*) включає n_1 однакових елементів, причому відмова будь-якого елемента виникає при його відмові в міжремонтний період або під час планових ремонтів, коли елемент визнають непридатним для подальшої експлуатації (відмовившим). Вихідними даними для розрахунку є: *N* - число типів (найменувань) замінних елементів що налічується механізмі; n_1 - кількість елементів кожного і-того типу що налічується механізмі; F_1 - ймовірність відмов кожного *i*-того типу елементів для даного механізму; c_1- вартість одного елемента відповідного типу; *V* - загальна надійність механізму.

Нехай ймовірність відмови елемента *i*-того типу обчислена на підставі обробки масиву даних про відмови в міжремонтний період *T* дорівнює F_1. Тоді ймовірність появи m_1 відмов елементів *i*-того типу за цей період може бути визначена на підставі біномного закону розподілу.

$$P_1(T) = C_{n_i}^{m_i} F_i^{m_i} (1 - F_i)^{n_i - m_i}$$

Так як m_i є випадковою величиною, то потреба в змінних елементах (запасних частинах) буде задоволена, якщо буде виконуватися умова: $m_i \leq m_{3\,i}$, де m_i число наявних (на складі) запасних частин *i*-того типу.

Імовірність здійснення цієї умови є функцією розподілу випадкової величини m_i і визначається за формулою:

$$P_i(m_i \leq m_{3i}) = \sum_{m_i=0}^{m3_i} C_{n_i}^{m_i} F_i^{m_i}(1-F_i)^{n_i-m_i}$$

Кількість запасних частин вважається достатнім, якщо за кожним замінним елементом виконується умова задоволеності запасних частин. Тоді для системи, що складається з елементів N різних типів, ймовірність здійснення умови достатності запасних частин, що є в наявності на складі матиме вигляд:

$$P_i[N(m_i \leq m_{3i})] = \prod_{i=1}^{N} \sum_{m_i=0}^{m_{3i}} C_{n_i}^{m_i} F_i^{m_i}(1-F_{i\,_})^{n_i-m_i}$$

Величина цієї ймовірності характеризує достатність забезпечення системи запасними частинами. Для машин і агрегатів, які безпосередньо беруть участь в металургійних процесах (агломераційні машини, доменні печі, сталеплавильні агрегати, МНЛЗ, прокатні стани), мінімальне значення такої ймовірності приймають V = 0,9, для допоміжного обладнання та окремо розташованих машин V = 0,7 ... 0,8. Тоді необхідну кількість запасних частин визначається з умови: $P_i[N(m_i \leq m_{3i})] \geq V$.

Для рішення поставленого завдання використано метод домінуючих послідовностей. В якості початкового, вибирають вектор $\overline{m_3{}^0}$= (0; 0; … ;0), відповідний відсутності запасних частин. При цьому достатність забезпечення

системи запасними частинами буде дорівнює ймовірності безвідмовної роботи (надійності) замінних елементів.

На першому кроці припускають, що склад запасних частин включає один елемент *i*-того типу. Тоді приріст достатності після першого кроку складе:

$$\eta_{l1} = \frac{P(m_i \leq 1) - P(m_i = 0)}{P(m_i = 0($$

$$= \frac{P(m_i \leq 1)}{P(m_i = 0)} - 1$$

Оптимальна кількість запасних частин може бути визначено з умови забезпеченості при мінімальній вартості. Задоволення споживачів за мінімальною вартістю виключає накопичення зайвих запасних частин, зменшує завантаження ремонтних цехів і знижує собівартість продукції, що випускається основними металургійними цехами.

Вартість запасних частин визначається з виразу:

$C_3 = \sum_{i=1}^{N} m_i c_i$, де c_i – вартість запасної частини *i*-того типу .

Сутність задачі зводиться до знаходження оптимального вектору кількісного складу запасних частин $\overline{m_3{}^0} = (m_{31}^0; m_{32}^0; \ldots; m_{3\,N}^0)$, при якому забезпечується мінімальне значення C_3 при виконанні обмеження достатності V забезпечення системи запасними частинами.

Тоді приріст достатності після першого кроку з урахуванням вартості c_i складе:

$$\eta_{l1} = \frac{1}{c_i}\left[\frac{P(m_i \leq 1)}{P(m_i = 0)} - 1\right]$$

З цього виразу визначають η_{l1} для кожного елемента *i*-того типу, знаходять максимальне η_{l1max} для відповідного елемента. Таким чином, на першому етапі отримують домінуючий вектор для якого передбачено наявність на складі одного запасного елемента для елемента що

експлуатується в машині і має максимальне збільшення достатності.. Для інших елементів запас на першому кроці не передбачають.

На другому кроці перераховують приріст достатності по тому елементу запас якого передбачили:

$$\eta_{2i} = \frac{1}{c_i}\left[\frac{P(m_i \leq 2)}{P(m_i \leq 1)} - 1\right]$$

і порівнюють це збільшення зі збільшеннями, обчисленими на першому кроці для інших елементів. Знову вибирають максимальне збільшення достатності і в запас вводять елемент відповідний цьому збільшенню. Якщо виявляється, що знову обчислений приріст достатності η_{2i} максимальний в порівнянні з обчисленими приростами достатності для інших елементів на першому кроці, то необхідно в запас внести ще один елемент того ж типу. Отже, на другому кроці отримують домінуючий вектор, $\overline{m_3}$ в якому вже або два елементи зберігаються на складі, будуть дорівнюють одиниці, або один елемент буде перебувати на складі в двох примірниках (в залежності від збільшення достатності), наявність інших на складі дорівнюватиме 0.

На довільному k-тому кроці оптимізації приріст достатності забезпечення запасними частинами складе:

$$\eta_{ki} = \frac{P(m_i \leq 2)}{P(m_i \leq 1)} - 1$$

Або з урахуванням вартості елемента c_i :

$$\eta_{ki} = \frac{1}{c_i}\left[\frac{P(m_i \leq m_{ki})}{P(m_i \leq m_{(ki)_i})} - 1\right]$$

З цього виразу знаходять $\eta_{ki\,max}$і визначають домінуючий вектор $\overline{m_3}$.

На кожному кроці перевіряють достатність забезпечення запасними частинами і порівнюють із заданою V. Завдання вважається вирішеним, якщо виконується умова забезпеченості запасними частинами.

Отже певно за допомогою вектора $\overline{m_3{}^0}$ склад запасних частин повинен поповняться перед кожним черговим ремонтом машини. Виходячи з цього, визначають необхідну кількість запасних частин на планований календарний період (місяць, квартал, рік). Такий розрахунок потребує певних знань і навичок. Може займати досить багато часу, особливо для великих і складних механізмів. А через людський фактор обчислення можуть мати деяку похибку.[13, 14]

1.2.2 Альтернативні способи розрахунку

Другий спосіб — це розрахунок за допомогою табличного процесора, наприклад MS Excel, де можна створити відповідні таблиці, додати потрібні формули для розрахунку, або використати макроси чи створити потрібний VBA код. Використання табличних процесорів, таких як Microsoft Excel, для розрахунку потреби в запасних частинах є поширеною практикою на багатьох підприємствах. Це дозволяє автоматизувати рутинні розрахунки, підвищити точність даних та отримати наочну візуалізацію результатів. Даний вид розрахунку значно швидший за попередній, але теж має свої недоліки, такі як: досить тривала підготовка до розрахунку, неповна автоматизація, а також потреба у певних знаннях для використання програми. Окрім того, виділяють ряд обмежень на використання табличного процесора, а саме: для дуже складних моделей може бути важко створити і підтримувати розрахунки в Excel, для великих обсягів даних розрахунки можуть займати багато часу; існує ризик виникнення помилок при введенні формул та даних.

Отже, Excel є потужним інструментом для розрахунку потреби в запасних частинах. Він дозволяє створювати гнучкі та наочні моделі, автоматизувати рутинні операції та аналізувати великі обсяги даних. Однак,

для складних проектів може знадобитися використання більш спеціалізованого програмного забезпечення.

Інший спосіб передбачає використання статистичних методів в програмному забезпеченні, наприклад, IBM SPSS Statistics, для аналізу історичних даних та прогнозування майбутнього попиту. Програма дозволяє легко імпортувати дані з різних джерел, очищати їх, трансформувати та візуалізувати, що є важливим для підготовки даних до аналізу. SPSS дозволяє створювати багатофакторні моделі, що враховують вплив різних змінних на попит на запасні частини (наприклад, сезонність, тренди, вплив інших товарів). Багато рутинних операцій можна автоматизувати, наприклад на етапі розрахунку отриманої надійності механізму, що економить час і зменшує ризик помилок. Хоча IBM SPSS Statistics є потужним інструментом для статистичного аналізу даних, він має певні обмеження та недоліки, особливо при його використанні для прогнозування. Ліцензія на SPSS може бути досить дорогою, особливо для невеликих компаній або окремих користувачів. Інтерфейс SPSS може бути складним для новачків, особливо для тих, хто не має досвіду роботи зі статистичними програмами. Вивчення всіх функцій і можливостей програми вимагає часу і зусиль. Хоча SPSS дозволяє будувати складні моделі, для деяких дуже специфічних або нестандартних задач може знадобитися додаткове програмування або використання інших інструментів. (Рис. 1)

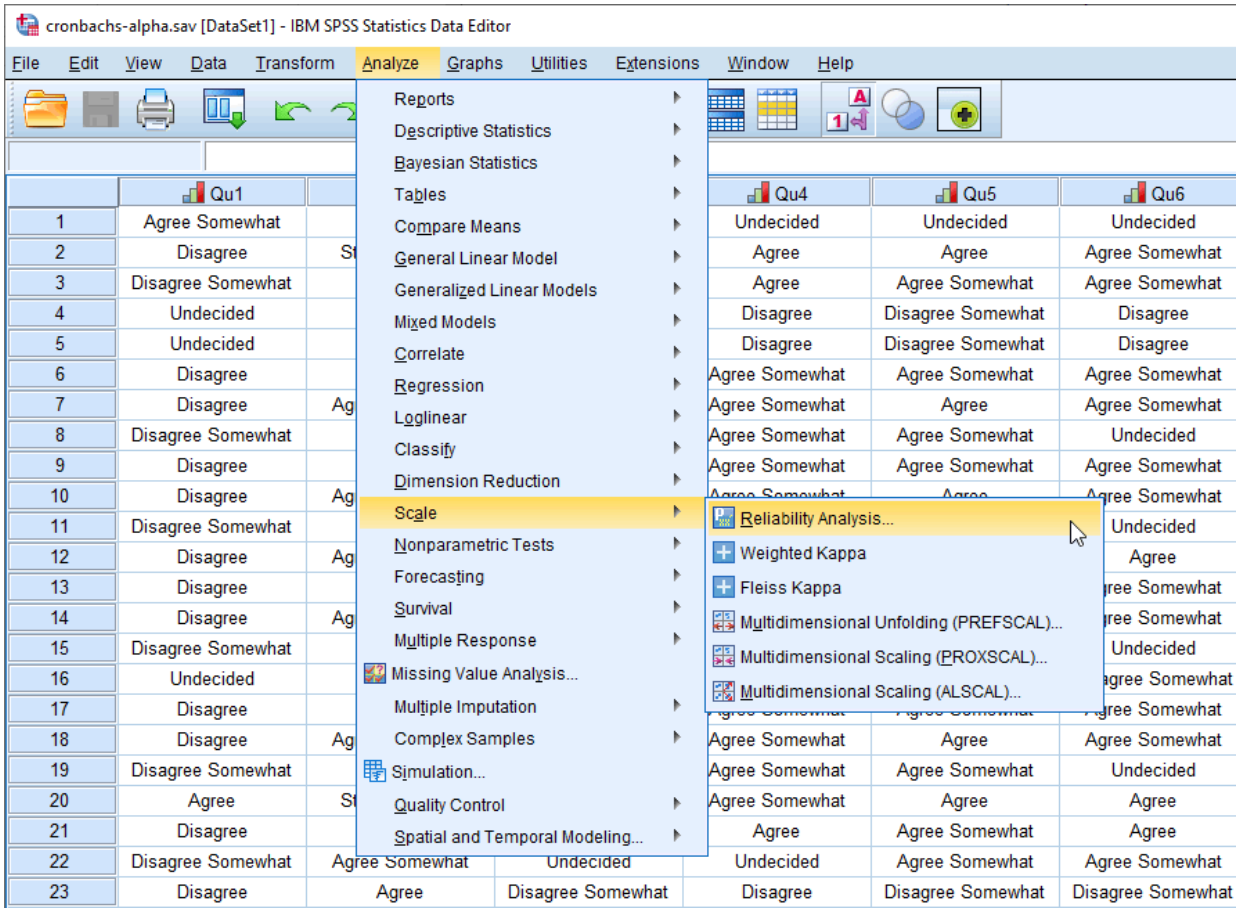


Рисунок *1 — Інтерфейс додатку «SPSS Statistics»*

Альтернативним способом рішення проблеми є створення власного програмного додатку для прогнозування. Це найгнучкіший спосіб вирішення задачі, який дозволяє повністю контролювати процес і створити рішення, яке ідеально відповідає вашим конкретним потребам. Перевагами його використання є висока швидкість розрахунків, що не залежить від кількості введених елементів. Додаток має просте та інтуїтивно зрозуміле управління, навіть користувач без спеціальних знань зможе легко розібратися в інтерфейсі та налаштувати програму під свої потреби. Інтерфейс використовує зрозумілу мову, що дозволяє уникнути зайвої складності. Графіки, діаграми та таблиці наочно демонструють результати розрахунків, що полегшує аналіз інформації. Користувач додатку має можливість проводити розрахунки, враховуючи вартість елементів або ні.

1.2.3 Спеціалізовані програмні системи

Застосунок Ansys є досить популярною системою аналізу, використовується фахівцями в області механіки, будівництва, автоматизації, тощо. Призначена для вирішення різних інженерних задач, розрахунків, аналізу і моделювання фізичних процесів, вирішення просторових завдань, механіки деформації твердого тіла та будівельної механіки(Рис. 2).

ANSYS може використовуватися для прогнозування довговічності конструкцій під впливом різних навантажень, температур та інших факторів. Це дозволяє оцінити ризик виходу з ладу конструкцій та вжити заходів для його запобігання.

ANSYS може використовуватися для виявлення слабких місць у конструкціях, де концентрація напружень є найвищою. Це дозволяє змінити конструкцію таким чином, щоб підвищити її міцність та довговічність.

ANSYS може використовуватися для оцінки впливу дефектів (тріщин, пор, включень) на міцність та довговічність конструкцій. Це дозволяє визначити, чи є дефект критичним і чи потребує він ремонту.

ANSYS сам по собі не розраховує потребу в запасних частинах. Він надає дані про напруження, деформації, довговічність, які потім використовуються в інших моделях для прогнозування відмов і розрахунку потреби в запчастинах.

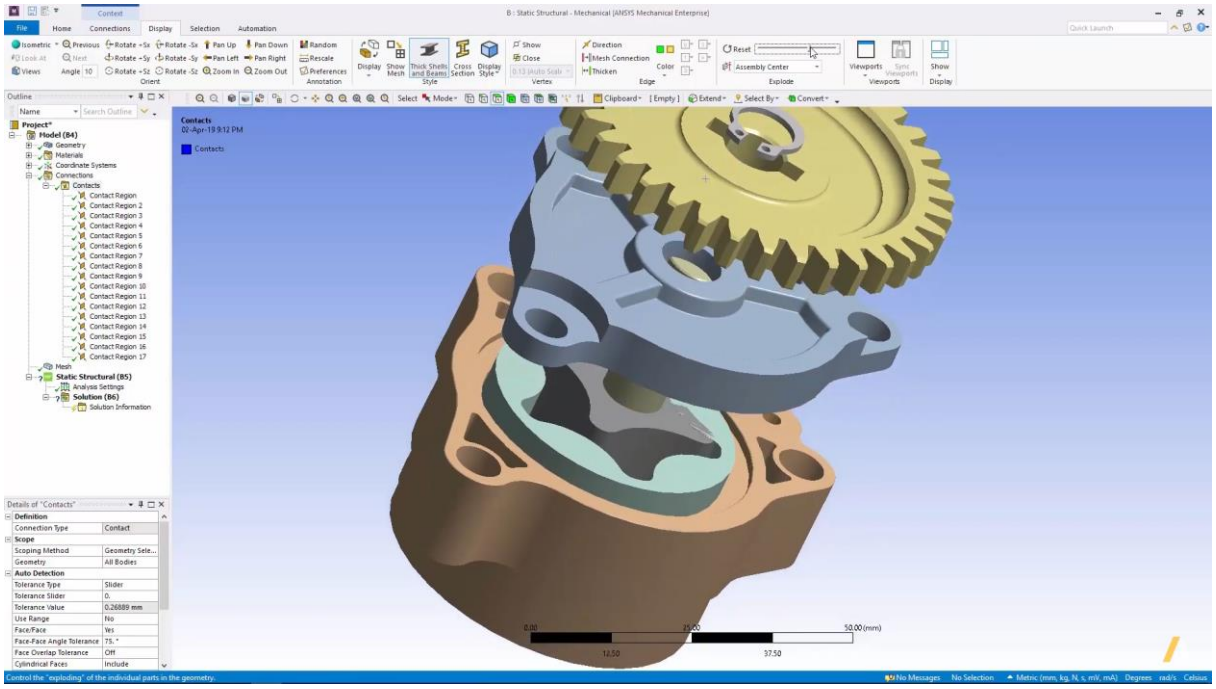


Рисунок *2 — Інтерфейс додатку «ANSYS»*

Основними перевагами використання ANSYS є: висока точність та надійність результатів, широкий спектр можливостей, доступність інтеграції з іншими CAD/CAM системами. Натомість через те що ANSYS є комерційним програмним забезпеченням — має високу вартість. Також являється досить складною програмою, яка потребує спеціальних знань та навичок для її використання. До того ж потребує потужне обчислювальне обладнання.

Веб-застосунок «Critical Spare Parts Calculator» призначений для управління запасними частинами в контексті технічного обслуговування виробничих підприємств. Основним функціоналом програмного продукту є: розрахунок витрат на запасні частини, рекомендації щодо закупівлі, оптимізація складу (див. Рис.3, 4).

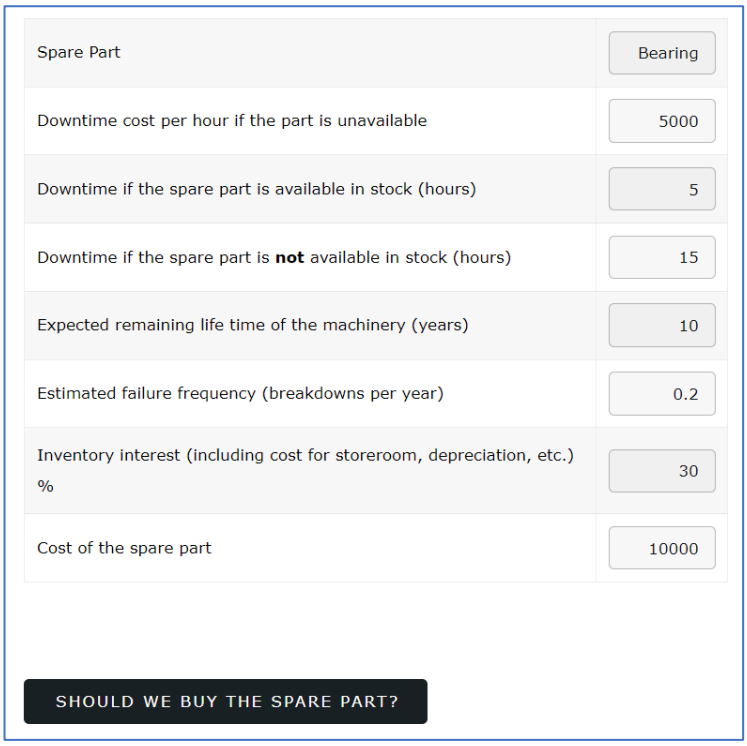


Рисунок *3 — Інтерфейс додатку «Critical Spare Parts Calculator» (введення даних)*

Завдяки продуманому алгоритму та інтуїтивному інтерфейсу, додаток допомагає:

- точно прогнозувати потребу в запасних частинах на основі історичних даних та аналізу трендів;
- зменшити витрати на зберігання та уникнути простоїв через відсутність необхідних запасних частин;
- автоматизувати розрахунки та надає рекомендації, що спрощує процес прийняття рішень;
- зменшити простої та оптимізувати запаси, що сприяє підвищенню загальної продуктивності підприємства;
- підвищити ефективність технічного обслуговування, завдяки автоматизації рутинних завдань, співробітники можуть зосередитися на більш складних завданнях, що підвищує загальну ефективність роботи підприємства.

Серед недоліків веб-додатку можна виділити залежність від точності даних та його обмежену гнучкість.



Extra cost per breakdown without the spare part in stock (C)	50000
Probability that the spare part will be used (P)	86 %
Expected downtime cost without the spare part (=C*P)	**43233**
Expected holding cost of the spare part (H)	17030
Expected obsolescence cost if the spare part never will be used (O)	1353
Total spare part costs (=H+O)	**18383**

YOU SHOULD PROBABLY **BUY** THE SPARE PART! - DOWNTIME COSTS ARE HIGHER THAN SPARE PARTS COST

Рисунок *4 — Інтерфейс додатку «Critical Spare Parts Calculator» (результат розрахунків)*

Веб-застосунок «Spares Calculator» призначений для оптимізації управління запасними частинами та зменшення ризику їх відсутності. Основним функціоналом програмного продукту є: розрахунок ризику відсутності запасних частин; оптимізація рівнів запасів; інтерпретація даних у вигляді графіків та таблиць; генерація звітів; аналіз повільно рухомих запасів (див. Рис. 5).

Перевагами «Spares Calculator» є те що додаток:

- дозволяє точно визначити, скільки потрібно витратити та коли, що допомагає в управлінні бюджетом;
- завдяки точним даним та прогнозам, користувачі можуть краще планувати свої операції та уникати непередбачених простоїв.

Застосунок використовується клієнтами в більш ніж 30 країнах.

Недоліками ж даного веб-додатку є залежність від точності даних та обмеження моделей.

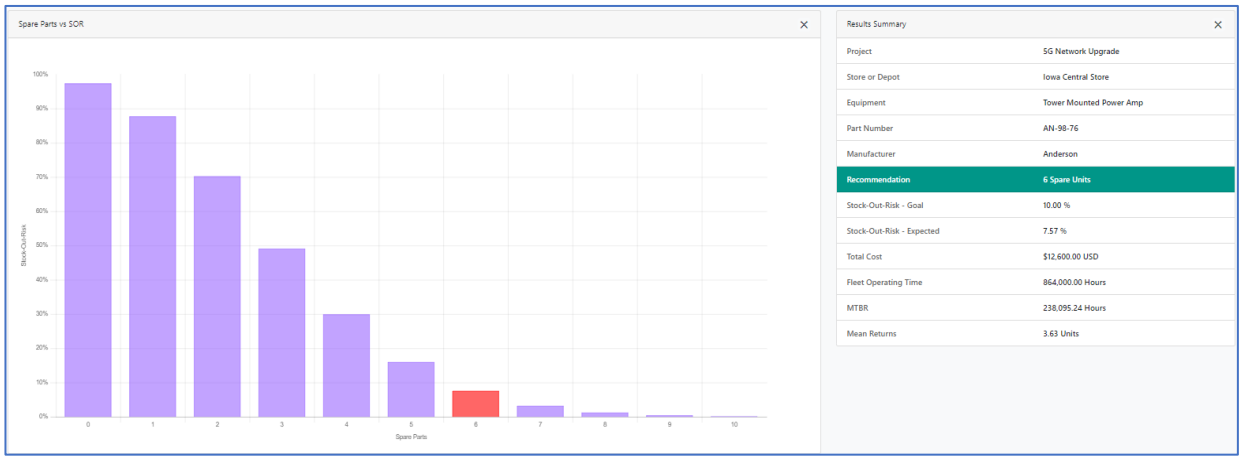


Рисунок *5 — Інтерфейс додатку «Spares Calculator»*

Тож такі інструменти є корисними для підприємств, які прагнуть оптимізувати свої процеси технічного обслуговування та зменшити витрати на запасні частини. Але у цих застосунках визначається економічна доцільність покупки змінних елементів. Під час дослідження проблеми все ж основним аспектом став розрахунок потреби у запасних частинах саме з точки зору надійності обладнання.

На основі дослідження аналогів програмного забезпечення для управління запасами на виробництві (таких як «Critical Spare Parts Calculator» та «Spares Calculator») виявлено ряд недоліків існуючих рішень. Для розробки більш ефективного та конкурентоспроможного програмного продукту було проведено порівняльний аналіз, результати якого представлені в таблиці 1.

Таблиця *1 — Проблематика яку вирішує даний додаток*

Проблема	Вирішення
Потрібно правильно підготуватися до капітального ремонту обладнання.	За допомогою додатку розраховуємо потрібну кількість запасних частин до відповідного обладнання, тоді можна зробити замовлення на запчастини і вчасно їх доставити.
Надлишок запчастин певного типу що не використовуються у таких кількостях.	Додаток допоможе порахувати необхідну їх кількість, відповідно визначити яку кількість деталей замовляти щоб не утворювався залишок.
Нестача елементів виявлена під час виконання ремонту	Завчасно провівши розрахунки у застосунку можна уникнути такої ситуації замовивши потрібну кількість кожного типу запасних частин завчасно.
Зберігання великої кількості елементів на складі	Обчисливши точну потребу і виконуючи закупки відповідно до розрахунків, така проблема не виникатиме
Розрахунок потреби у запасних частинах під час проведення капітального ремонту досить складний і об’ємний	Для використання додатку не потрібно володіти методикою проведення такого розрахунку, вводимо дані і отримуємо результат
Зменшення надійності обладнання	Додаток дозволяє задати мінімальну потрібну надійність обладнання, після проведення усіх обчислень отримати надійність з урахуванням заміни визначених елементів що буде більшою за задану на початку.
Врахування вартості змінних елементів	Завдяки наявності двох модулів можна обрати враховувати вартість елементів при розрахунках чи ні.
Визначення кінцевої суми для закупки	Після проведення обчислень, у разі вибору модуля що враховує вартість запасних частин під час розрахунку , у віконечку з результатами отримаємо вартість закупу кожного типу змінних елементів та загальну вартість по обчислюваному механізму.

Таблиця містить опис типових проблем, що виникають під час планування та виконання капітального ремонту, а також пропонує конкретні рішення, які може забезпечити нове програмне забезпечення.

1.3 Постановка завдання

Мета кваліфікаційної роботи — це створення програмного забезпечення для розрахунку потреби у запасних частинах. Постановка завдання дослідження включає наступні етапи:

1. Аналіз потреб і вимог: Визначення основних потреб та вимог, які повинно задовольняти розроблене програмне забезпечення для розрахунку потреби у запасних частинах. Зокрема проведення коректного розрахунку, можливість зберігання даних, та додавання файлів зображень, креслень, тощо. Наявність простого і зрозумілого інтерфейсу для легкої взаємодії.
2. Вибір технологій: Вибір технологічного стеку, який найкраще відповідає потребам дослідження. У даному випадку, вибір фреймворку Django, як основного для розробки системи.
3. Розробка архітектури: Визначення архітектури програмної системи, враховуючи можливість ведення розрахунків. Врахування принципів гнучкості, розширюваності та ефективності програмного продукту.
4. Розробка функціональності: Реалізація основних функціональних можливостей застосунку для розрахунку потреби у запасних частинах, враховуючи запис даних, ведення розрахунку, та одержання результатів. Врахування принципів оптимізації продуктивності.
5. Апробація: Виконання ряду перевірок для переконання в коректності роботи розробленого програмного продукту. Валідація додатку на основі визначених вимог та порівняння результатів з іншими існуючими системами проведення аналогічних розрахунків.

1.4 Висновки до розділу 1

1. Виконано огляд літературних джерел, де описано розрахунки для визначення потреби у запасних частинах.

2. Було оглянуто головні проблеми створення програмного забезпечення для ведення розрахунків: складність розробки, потреба у розумінні предметної області, знання і розуміння як працює розрахунок.
3. Проаналізовані існуючі рішення для визначення потреби у запасних частинах. Були визначені їх переваги та недоліки.
4. Визначена постановка завдання дослідження.

2 РЕАЛІЗАЦІЯ ПРОГРАМНОГО РІШЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ЗАПАСІВ

2.1 Вимоги до програмного продукту

Функціональні вимоги

- Авторизація та автентифікація:
  - Реєстрація нових користувачів з підтвердженням електронної пошти.
  - Вхід в систему за допомогою логіну та паролю.
  - Розподіл ролей користувачів.
- **Управління даними про обладнання:**
  - Додавання, редагування та видалення інформації про обладнання.
  - Завантаження технічної документації для обладнання (кресленики, специфікації).
  - Додавання вузлів до машин.
- **Управління даними про запасні частини:**
  - Створення каталогу запасних частин з детальною інформацією (назва, номер кресленика, номер по специфікації, кількість, вартість, ймовірність відмови).
  - Завантаження зображень/креслеників запасних частин.
  - Зв'язок запасних частин з конкретними вузлами обладнання.
- **Розрахунок потреби в запасних частинах:**
  - Вибір методу прогнозування (з врахуванням вартості запасних частин та без врахування вартості запасних частин).
  - Відображення результатів розрахунку у вигляді таблиць.
  - Відображення вартості закупки
- **Пошук та фільтрація:**
  - Швидкий пошук обладнання та запасних частин за різними критеріями.
  - Фільтрація даних за різними параметрами.

Вимоги до інтерфейсу користувача (UI/UX)

- Інтуїтивно зрозумілий та зручний інтерфейс.
- Адаптивний дизайн (підтримка різних розмірів екранів).
- Використання сучасних UI-компонентів.
- Можливість налаштування інтерфейсу під потреби користувача.
- Валідація введених даних для запобігання помилкам.

Нефункціональні вимоги

- **Продуктивність:** Швидкий час відгуку на дії користувача, мінімальний час завантаження сторінок.
- **Надійність:** Безперебійна робота застосунку, мінімізація ризику втрати даних.
- **Безпека:** Захист від несанкціонованого доступу, шифрування даних.
- **Масштабованість:** Можливість збільшення кількості користувачів та обсягу даних без втрати продуктивності.

2.2 Проєктування системи

В процесі проектування архітектури системи для визначення потреб та логіки її роботи сформовано UML діаграми [11]. Діаграма активностей представлена на рисунку 6 візуалізує принцип роботи розробленого застосунку. Спочатку користувач вносить необхідні дані, під час розрахунку будується домінуючий вектор прирощення достатоності, і порівнюється з мінімальною надійністю вузла, що була задана при його створені. Розрахунок проводиться циклічно до моменту, поки отримана надійність перевищить задану.

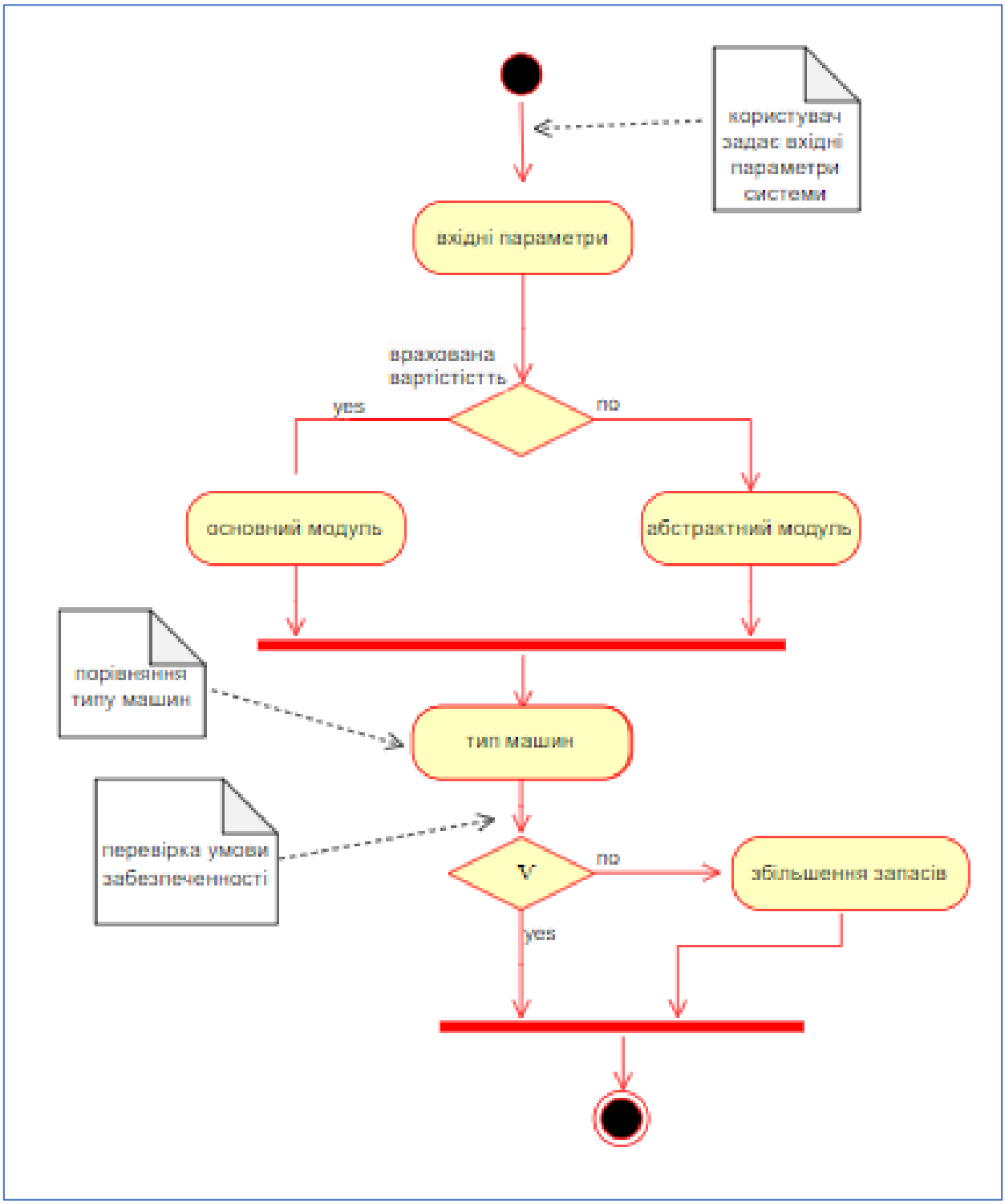


Рисунок *6 — Діаграма активностей*

Наведена на рисунку 7 діаграма Use Case представляє взаємодію користувача з системою. Вона описує систему, яка дозволяє користувачеві виконувати розрахунки, пов'язані з плануванням запасів деталей. Система гнучка і дозволяє користувачеві задавати різні параметри для отримання

необхідних результатів. розрахувати необхідну кількість деталей з урахуванням різних факторів, таких як вартість, надійність та ймовірність відмови. Для демонстрації того, як саме користувач працює з системою представлено діаграму послідовності дій користувача (див. Рис. 8). У таблиці 2 представлено опис прецедентів для сценаріїв дій користувача

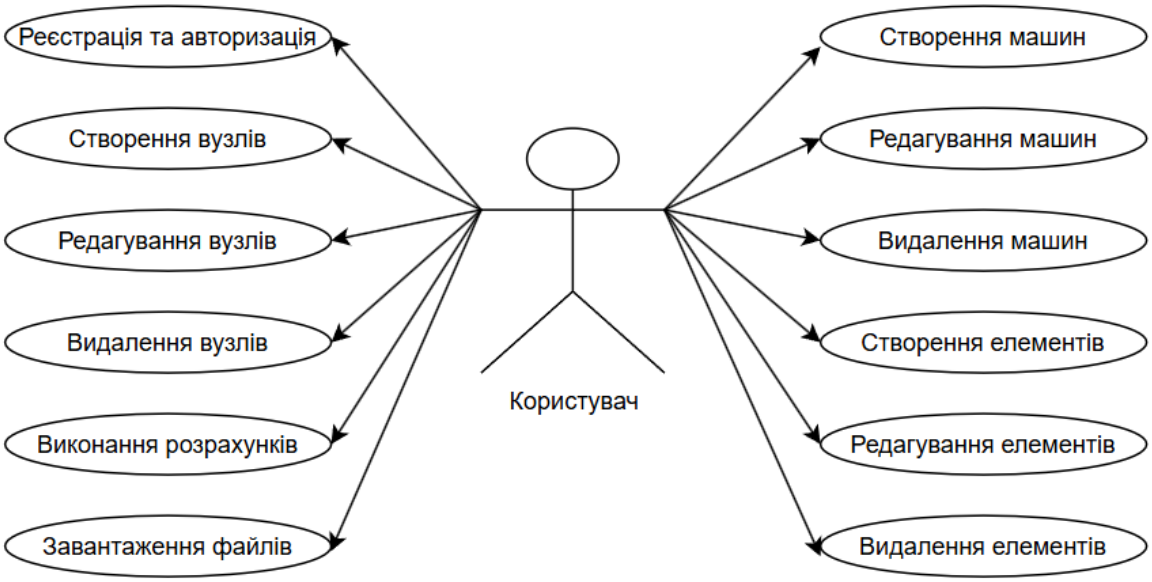


Рисунок *7 — Use Case діаграма додатку*

Таблиця *2 — Опис прецедентів*

Прецедент	**Актор**	**Сценарій**	**Підсумкові умови**
1	**2**	**3**	**4**
Вхід в систему	Зареєстрований користувач	Зареєстрований користувач вводить дійсні логін і пароль у відповідних полях для вводу	Користувач авторизувався і бачить дані системи
Створення машини	Авторизований користувач	Авторизований користувач натискає кнопку створити машину, додає номер кресленика, існуючі вузли та файл з даними	На головній сторінці створилася нова машина, запис зберігся до бази даних
Створення вузла	Авторизований користувач	Авторизований користувач натискає кнопку створити вузол, вказує номер кресленика та мінімальну потрібну надійність вузла, додає файл з даними	На сторінці створився новий вузол, запис зберігся до бази даних
Додавання елементів	Авторизований користувач	Авторизований користувач натискає кнопку додати новий елемент, вказує кресленик вузла до якого належить даний елемент та номер по специфікації; додає назву, дані про кількість таких елементів, вартість та ймовірність відмови	На сторінці створився новий елемент, запис зберігся до бази даних
Редагування машини	Авторизований користувач	Авторизований користувач натискає кнопку редагувати машину, додає номер кресленика, існуючі вузли та файл з даними	На головній сторінці оновилася інформація про машину, запис зберігся до бази даних

1	2	3	4
Редагування даних про вузол	Авторизований користувач	Авторизований користувач натискає кнопку редагувати вузол, вказує номер кресленика та мінімальну потрібну надійність вузла, додає файл з даними	На сторінці оновилася інформація про редагований вузол, запис зберігся до бази даних
Редагування даних про елемент	Авторизований користувач	Авторизований користувач натискає кнопку редагувати елемент, вказує кресленик вузла до якого належить даний елемент та номер по специфікації; додає назву, дані про кількість таких елементів, вартість та ймовірність відмови	На сторінці оновлюються дані про редагований елемент елемент, запис зберігся до бази даних
Проведення розрахунку	Авторизований користувач	Натискає на кнопку розрахувати біля потрібного вузла	Отримує таблицю з результатами розрахунку

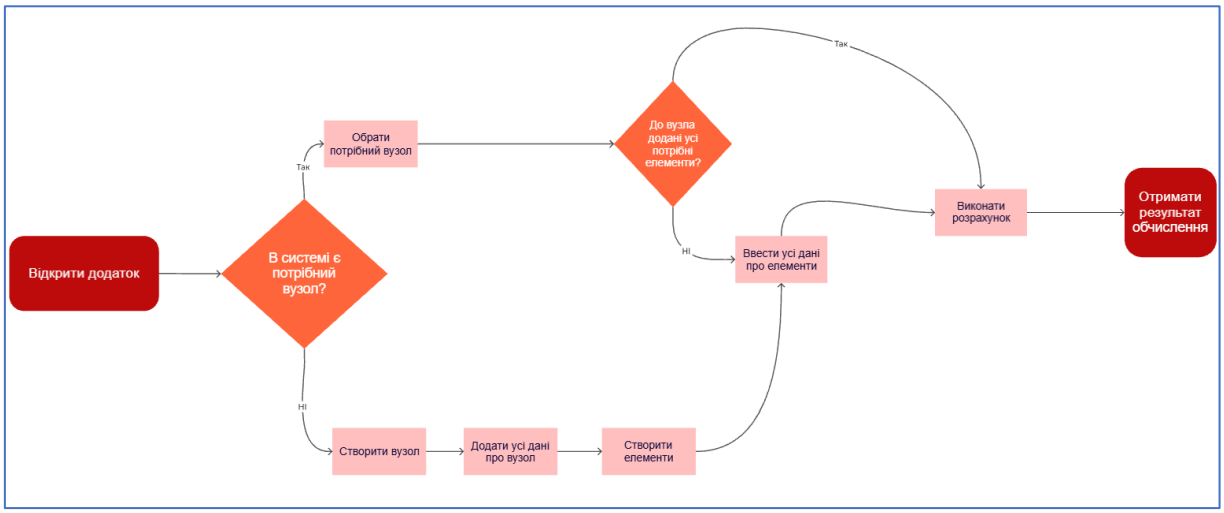


Рисунок *8 — Послідовність дій користувача*

2.3 Архітектура додатку

Для створення надійного та гнучкого веб-застосунку, який допомагатиме підприємству ефективно керувати запасами запасних частин, використано архітектуру, яку пропонує фреймворк Django. Цей підхід, відомий як MTV (Model-Template-View), є варіацією класичного MVC і слугує чудовим фундаментом для організації коду, забезпечуючи чіткий поділ обов'язків між різними компонентами системи [8, 9].

Основою даного застосунку є Модель (Model). Це цифровий фундамент, що відповідає за всі дані та їхню структуру. Моделі описують суть даних: що таке "Запасна частина" з її характеристиками (номер кресленика вузла до якого належить даний елемент та його номер по специфікації; назва, дані про кількість таких елементів, вартість та ймовірність їх відмови), що таке "Вузол", до якої ця частина може знадобитися, та "Машина" до якої цей вузол відноситься. Моделі не лише визначають структуру даних, але й забезпечують взаємодію з базою даних усієї цієї критично важливої інформації.

Але самі по собі дані не виконують розрахунків і не взаємодіють з користувачем. Для цього використовується Представлення (View). В цій частині даного застосунку, міститься вся бізнес-логіка. Саме View обробляє запити користувача, наприклад, коли інженер просить розрахувати потребу в запчастинах для певного вузла на наступний капітальний ремонт. View звертається до Моделей, щоб отримати необхідні дані, виконує складні алгоритми розрахунку потреби у запасних частинах для виконання капітального ремонту обладнання, враховуючи різні фактори, і готує результат для показу. View виступає як координатор, що зв'язує дані та їхню обробку з кінцевим представленням.

Коли Представлення (View) підготувало дані та результати розрахунків, їх потрібно представити користувачеві у зрозумілому та зручному вигляді. За це відповідають Шаблони (Template). Це front-end частина застосунку — HTML-файли, які визначають, як саме виглядатиме інтерфейс: таблиці зі

списками запчастин, форми для введення даних, інформативні звіти з результатами розрахунків, тощо. Шаблони отримують дані від View і вставляють їх у заздалегідь визначені місця, формуючи кінцеву веб-сторінку, яку бачить користувач. Вони зосереджені виключно на візуальному представленні, не втручаючись у логіку чи роботу з даними.

Коли користувач надсилає запит (наприклад, натискає кнопку "Розрахувати"). Django спрямовує цей запит до відповідного View. View звертається до Моделей за даними, проводить обчислення, обирає потрібний Шаблон і передає йому оброблену інформацію. Шаблон генерує HTML-сторінку з результатами, яка й відправляється назад користувачеві.

Такий чіткий поділ відповідальності між Моделлю (дані), Шаблоном (вигляд) та Представленням (логіка) робить даний застосунок для розрахунку запчастин значно простішим в розробці, тестуванні та подальшій підтримці. За необхідності є можливість змінювати алгоритми розрахунку (у View) чи дизайн інтерфейсу (у Template), мінімально зачіпаючи інші частини системи. Це сприяє гнучкості, масштабованості та полегшує роботу над проєктом. Таким чином, архітектура MTV у Django стає надійною основою для побудови ефективного та керованого веб-рішення.

2.4 Засоби та технології розробки програмного забезпечення

Для розробки програмного забезпечення для розрахунку потреби у запасних частинах під час проведення капітального ремонту обладнання, використовуються різноманітні технології, а саме:

Python

Створення ефективного веб-застосунку для розрахунку потреби у запасних частинах, особливо в умовах планування та проведення складних капітальних ремонтів, вимагає надійного, гнучкого та продуктивного технологічного рішення. Саме тому було обрано поєднання мови програмування Python та високорівневого веб-фреймворку Django, що

демонструє свої найкращі сторони, забезпечуючи міцний фундамент для розробки.

Вибір Python як основи для такого проєкту є виправданим, оскільки ця мова здобула світову популярність не лише завдяки своїй відносній простоті синтаксису, що значно прискорює процес розробки та полегшує розуміння коду навіть у складних проєктах, але й завдяки своїй потужності та універсальності. Для застосунку що допоможе провести розрахунки потреби у запасних частинах для проведення капітального ремонту обладнання, де потрібні не лише стандартні веб-функції, а й складні розрахунки та аналіз даних, Python пропонує величезну екосистему бібліотек. Інструменти для аналізу даних (як NumPy) що легко інтегруються, дозволяючи реалізувати просунуті алгоритми для визначення потреби в запчастинах. Швидкість розробки, яку забезпечує Python, дозволяє швидше реагувати на потреби бізнесу та впроваджувати нові функції, а велика та активна спільнота розробників гарантує доступ до знань, готових рішень та підтримки.

Django

Django — це потужний набір інструментів (фреймворк), написаний мовою програмування Python, який допомагає швидко та легко створювати веб-сайти та веб-додатки. Він є безкоштовним і має відкритий код, тобто будь-хто може його вільно використовувати та модифікувати.

Ключова перевага Django полягає в тому, що він значно спрощує процес розробки. Створюючи будь-який веб-сайт, розробник неминуче стикається з однаковими завданнями — потрібно реалізувати вхід та реєстрацію користувачів, зробити зручну панель для керування вмістом сайту, налаштувати обробку даних, які надсилають користувачі через форми, організувати завантаження файлів тощо. Django надає вже готові, надійні компоненти для вирішення цих типових задач. Що дозволяє не витрачати час на створення вже існуючого стандартного функціоналу, а зосередитись на унікальних функціях проекту.

Маючи на меті створити систему для розрахунку потреби у запасних частинах для проведення капітальних ремонтів обладнання, Django є ефективним рішенням. Оскільки такий додаток буде оперувати значними обсягами даних: переліки обладнання, тисячі найменувань запчастин, їхні параметри, кресленики, тощо. Django має вбудовані потужні інструменти (зокрема, ORM — Object-Relational Mapper) для ефективної роботи з базами даних. Це суттєво спрощує зберігання, пошук, фільтрацію та аналіз великих масивів інформації. Також Django пропонує готову адміністративну панель. Тож під час розробки немає потреби з нуля розробляти інтерфейс для керування всіма цими даними — додавання нових позицій запчастин, редагування інформації про обладнання, тощо. Базовий функціонал адміністрування доступний майже одразу, що дозволяє швидко запустити додаток і зосередитись на розробці специфічної бізнес-логіки — алгоритмів розрахунку потреби у запасних частинах. Крім того, гнучка система автентифікації та авторизації Django дозволяє легко налаштувати різні рівні доступу для співробітників (наприклад, інженерів, менеджерів з логістики, керівництва) [1, 2].

SQLite

SQLite — це компактна та ефективна система керування базами даних, яка вирізняється своєю простотою та унікальним підходом. SQLite підтримує більшість стандартних команд мови запитів SQL. Вона дозволяє гнучко визначати типи даних для інформації — чи то цілі числа, текст, дійсні числа, чи бінарні дані, забезпечуючи їх точне зберігання. Для пришвидшення доступу до даних SQLite дозволяє створювати індекси, що значно підвищує продуктивність пошуку. Крім того, вона підтримує тригери (автоматичні дії у відповідь на зміни даних) та представлення (спеціальні збережені запити для спрощення доступу до певних даних).

Що робить SQLite справді особливою, так це її архітектура. На відміну від більшості баз даних, їй не потрібен окремий серверний процес для роботи. Вся база даних — усі таблиці, індекси та самі дані —— зберігається в одному-

єдиному файлі на комп'ютері чи пристрої, це робить її портативною та легкою. Що дає змогу легко скопіювати, перемістити чи зробити резервну копію всієї бази даних, просто працюючи з одним файлом.

Такий підхід дає низку переваг. По-перше, SQLite проста у використанні, розгортанні та управлінні. Немає потреби у складних налаштуваннях чи постійному адмініструванні сервера, що робить її ідеальним вибором для невеликих проектів, мобільних додатків, вбудованих систем або просто для швидкого старту розробки. По-друге, вона дуже ефективна та споживає мінімум системних ресурсів, забезпечуючи при цьому високу продуктивність для багатьох типових завдань. Попри свою простоту, SQLite забезпечує надійність даних завдяки підтримці ACID-транзакцій (атомарність, узгодженість, ізоляція, довговічність).

SQLite широко використовується у світі — у багатьох веб-браузерах, операційних системах, мобільних телефонах та безлічі інших програм. Вона є безкоштовною, з відкритим вихідним кодом, що дозволяє вільно використовувати її навіть у комерційних проектах без жодних ліцензійних платежів. До того ж, вона кросплатформна, тобто працює на багатьох операційних системах і підтримується великою кількістю мов програмування, що робить її гнучким та універсальним інструментом для розробників. [10]

HTML

HTML (HyperText Markup Language) є основною мовою розмітки, що використовується для створення веб-сторінок. Вона визначає структуру та семантику веб-документа, включаючи текстовий зміст, зображення, посилання, таблиці, форми і багато іншого. HTML вважається фундаментальною технологією для веброзробки.

Основні риси HTML:

Теги: HTML використовує теги для визначення елементів на веб-сторінці. Теги мають початковий та заключний тег, які вказують початок та кінець елемента. Наприклад, це параграф визначає параграф на сторінці.

Семантика: HTML надає широкий набір семантичних елементів, які допомагають визначати типи контенту на веб-сторінці. Наприклад, <header>, <nav>, <section>, <article>, <footer> - це семантичні елементи, які допомагають структурувати сторінку та розуміти її зміст.

Атрибути: HTML елементи можуть мати атрибути, які надають додаткові властивості або значення. Наприклад, атрибут "src" визначає шлях до зображення, а атрибут "href" визначає посилання на іншу сторінку.

Вбудовування мультимедіа: HTML дозволяє вставляти різноманітний мультимедійний контент, такий як зображення, відео або аудіо на веб-сторінки. Це здійснюється за допомогою відповідних елементів, таких як <img>, <video>, <audio>.

Форми: HTML надає можливість створювати форми, які дозволяють користувачам взаємодіяти з веб-сторінкою, вводити дані та надсилати їх на сервер. Елементи форми включають <input>, <textarea>, <select> та багато інших.

HTML є важливим елементом веб-розробки, оскільки вона визначає структуру та семантику веб-сторінок. Розуміння HTML є необхідним для будь-якого веб-розробника, оскільки вона є основою для подальшого вивчення .[19]

CSS

Cascading Style Sheets, який використовується для надання стилів та взаємодії з веб-сторінками.

CSS (каскадні таблиці стилів) — це мова стилів, яка використовується для розробки та візуального оформлення веб-документів, написаних мовою HTML.

CSS визначає зовнішній вигляд елементів на веб-сторінці: колір, шрифт, розмір, положення, інтервали та інші аспекти дизайну.

Основні можливості CSS:

Вибір селекторів: CSS використовує селектори для вибору елементів на веб-сторінці для застосування стилів. Селектори можуть вибирати елементи за типом, класом, ідентифікатором або іншими атрибутами. Властивості та

значення:CSS працює, встановлюючи властивості та їх значення для вибраних елементів. Наприклад, властивість «color» визначає колір тексту, а властивість «font-size» визначає розмір шрифту.

Розміщення та позиціонування: CSS надає спосіб контролювати розміщення елементів на сторінці. Такі властивості, як «margin», «padding» і «position», можна використовувати для впливу на положення та відступи елементів.

Каскадування та успадкування: CSS має концепцію каскадування, що означає, що стилі можуть бути визначені на різних рівнях (внутрішні, зовнішні таблиці стилів, атрибути стилю HTML тощо) і мають різні пріоритети. Успадкування дозволяє вам успадковувати стилі від батьківських елементів, що полегшує зміну стилів для багатьох елементів одночасно.

Анімації та переходи: CSS дозволяє створювати анімацію та переходи, які додають динаміку та руху вашим веб-сторінкам. Анімацію можна застосовувати до різних властивостей, таких як колір, розмір і положення.

CSS є невід’ємною частиною веб-розробки, оскільки дозволяє створювати привабливі та красиві веб-сторінки. CSS дозволяє відокремити вміст від презентації, покращуючи зручність обслуговування, змінність і контроль над дизайном сторінки. Це дозволяє розробникам творчо висловлювати свої ідеї та надавати своїм веб-документам безпечного та професійного вигляду [20].

2.4 Середовище розробки PyCharm

PyCharm, розроблене компанією JetBrains, являє собою інтегроване середовище розробки (IDE), призначене для мови програмування Python. Воно позиціонується як багатофункціональна платформа, що забезпечує комплексний інструментарій для всіх етапів життєвого циклу розробки програмного забезпечення: від написання та аналізу коду до налагодження, тестування та розгортання додатків.

Функціональним ядром PyCharm є інтелектуальний редактор коду, що забезпечує підсвічування синтаксису, автодоповнення на основі аналізу коду, статичний аналіз в реальному часі для виявлення помилок та потенційних проблем, а також засоби рефакторингу для поліпшення структури коду. Середовище інтегрує потужний графічний налагоджувач (debugger), який дозволяє покрокове виконання коду, інспекцію значень змінних та аналіз стеку викликів. Також наявна інтегрована підтримка фреймворків для модульного тестування (unit testing), що сприяє створенню надійного програмного забезпечення.

PyCharm надає інструменти для управління залежностями проекту, включаючи інтеграцію з менеджерами пакетів pip та conda, а також управління віртуальними середовищами. Інтегровані засоби для роботи з реляційними базами даних дозволяють встановлювати з'єднання, виконувати SQL-запити та маніпулювати даними безпосередньо з IDE. Глибока інтеграція з системами контролю версій (VCS), зокрема Git, забезпечує візуалізацію змін, управління гілками та історією комітів, що є критично важливим для командної розробки.

Особливої уваги заслуговує розширена підтримка веб-фреймворків у версії PyCharm Professional. Зокрема, реалізовано глибоку спеціалізовану інтеграцію з фреймворком Django, який використовується для розробки поточного додатку. Ця інтеграція включає такі функціональні можливості:

Надання шаблонів проектів для швидкої ініціалізації стандартної структури Django-додатку.

Пряма інтеграція з утилітою manage.py, що дозволяє виконувати стандартні команди Django (наприклад, runserver, makemigrations, migrate, shell) через графічний інтерфейс IDE.

Контекстно-залежне автодоповнення коду та навігація між різними компонентами Django-проекту: моделями (Models), представленнями (Views), URL-маршрутизаторами (URL patterns) та шаблонами (Templates).

Спеціалізовані можливості налагодження, що дозволяють встановлювати точки зупину та інспектувати змінні безпосередньо у коді Django-шаблонів.

Інтегрований доступ до інтерактивної оболонки Django Shell для виконання команд в контексті проекту.

Ця тісна інтеграція суттєво підвищує продуктивність розробки веб-додатків на базі Django. Крім того, PyCharm підтримує інтеграцію з іншими фреймворками (наприклад, Flask, FastAPI) та технологіями, такими як контейнеризація (Docker), системи безперервної інтеграції/доставки (CI/CD), хмарні платформи (AWS, Google Cloud, Azure) та інструменти для обробки великих даних.

До потенційних недоліків слід віднести відносно високі вимоги до системних ресурсів, особливо при роботі з великими проектами. Розширена функціональність, зокрема глибока підтримка веб-фреймворків (включаючи Django) та інструментів для роботи з базами даних, доступна переважно у комерційній версії Professional. Початківці можуть потребувати певного часу для освоєння багатофункціонального інтерфейсу середовища.

Таким чином, PyCharm є потужним та універсальним інструментальним засобом для розробки на Python. Його функціональність, особливо у версії Professional з розширеною підтримкою фреймворку Django, робить його ефективним вибором для професійних розробників, що працюють над складними проектами, зокрема у сфері веб-розробки, як-от створення додатка для розрахунку потреби у запасних частинах для проведення капітальних ремонтів обладнання [21].

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ

3.1 Проектування додатку

При проектуванні додатку відповідно до поставленої задачі та вимог до продукту було створено діаграму класів (Рисунок 9) та схему бази даних(Рисунок 10)

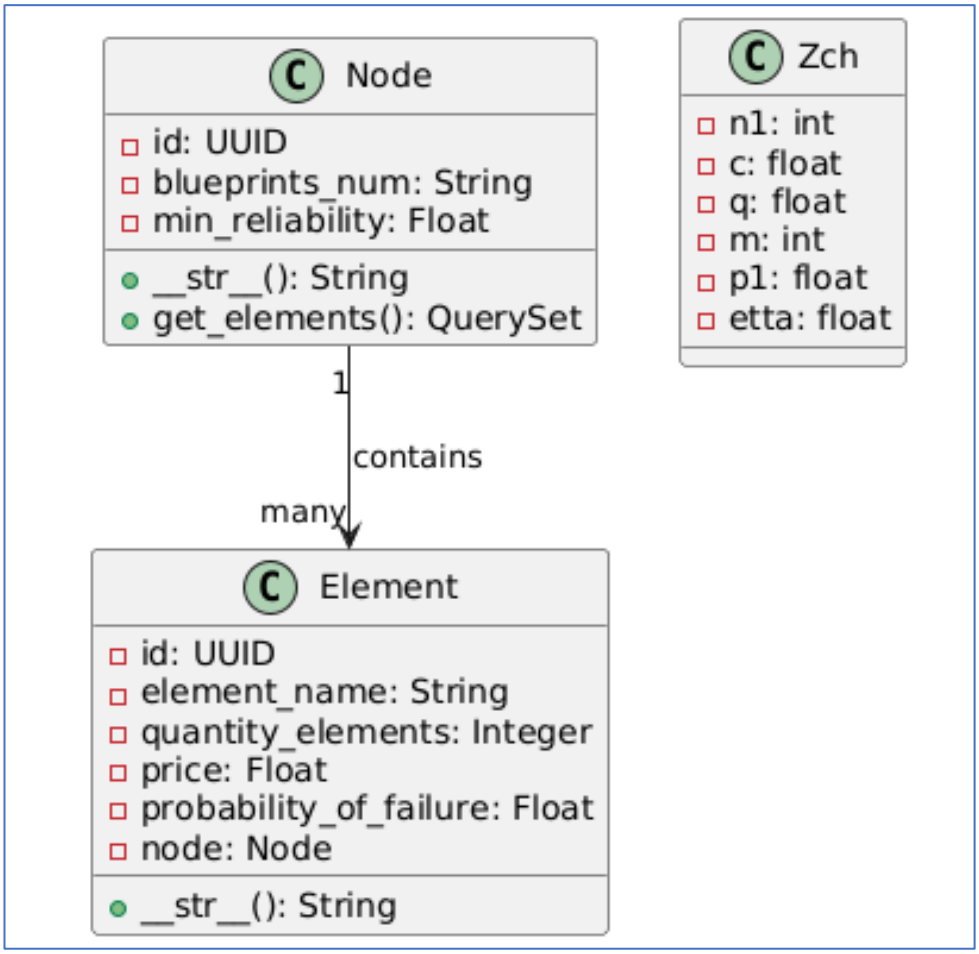


Рисунок *9 — Діаграма класів*

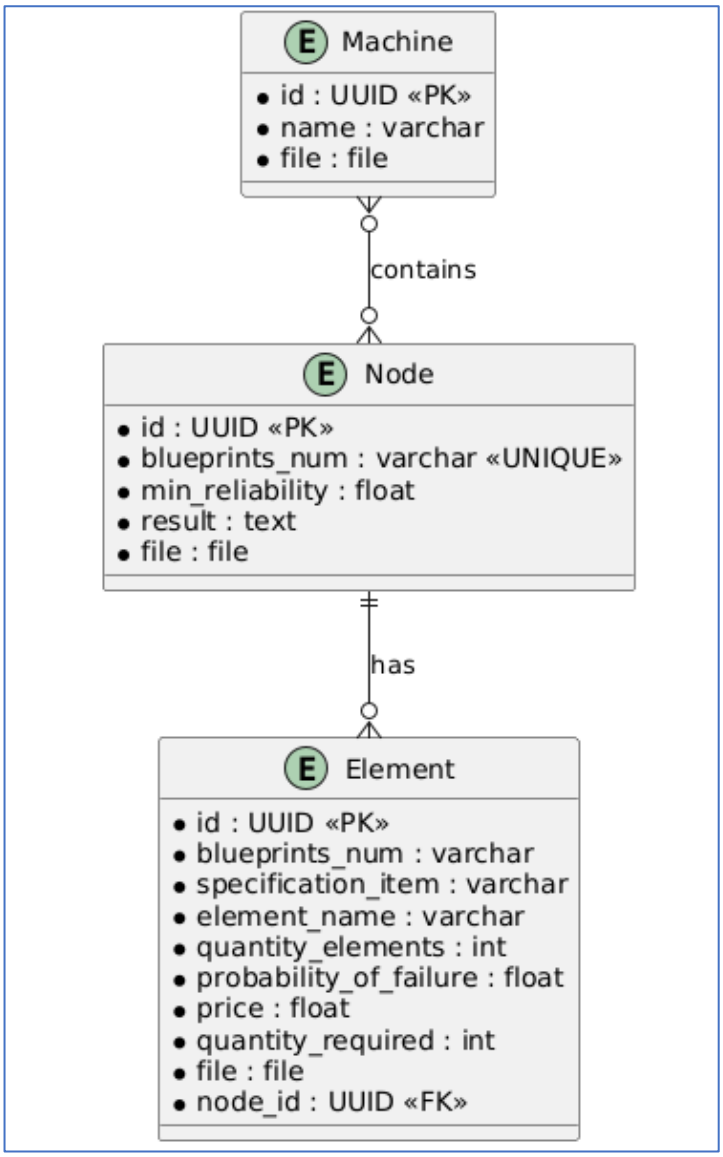


Рисунок *10 — Схема бази даних*

Програма призначена для оптимізації складу запасних частин шляхом розрахунку їх необхідної кількості для забезпечення заданого рівня надійності обладнання. Алгоритм враховує вартість елементів, ймовірність їх відмов та взаємозв'язки між ними.

3.2 Розробка програмного забезпечення

Додаток є простим та інтуїтивно зрозумілим для використання. Спочатку користувач має зареєструватися (Рис. 11), увійти в попередньо створений аккаунт (Рис. 12)

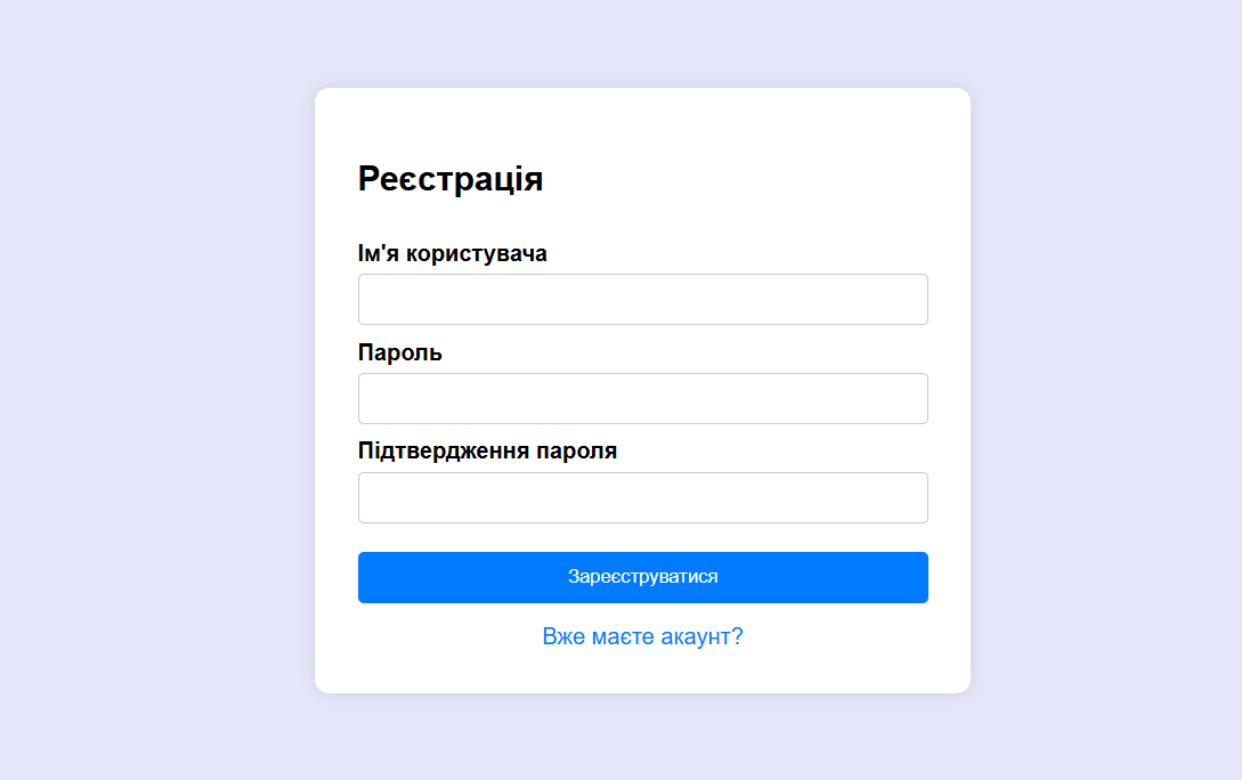


Рисунок *11 — Вікно реєстрації*

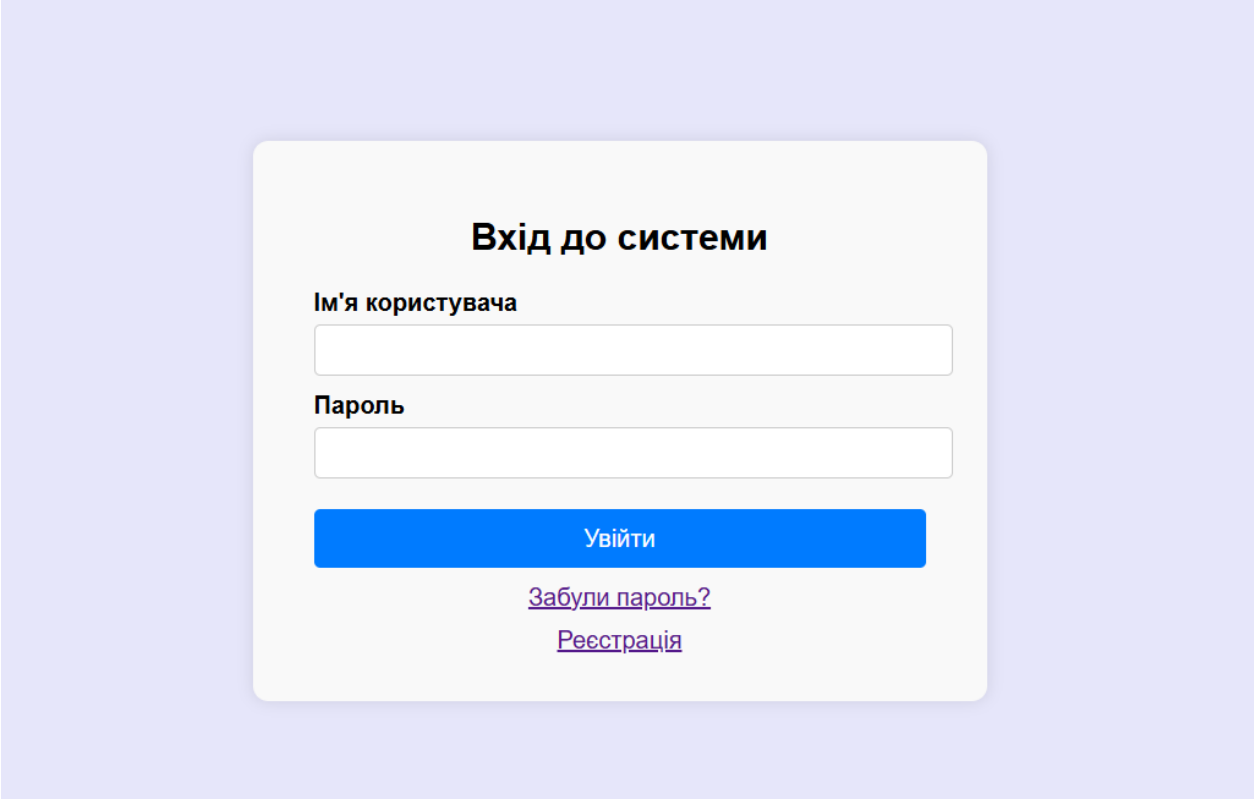


Рисунок *12 — Вхід до системи*

Після авторизації користувач опиняється на головній сторінці додатку (Рис. 13), де відображається список машин, де їх можна створювати (Рис. 14), змінювати (Рис. 15) та видаляти, що представлено на рисунку 16.

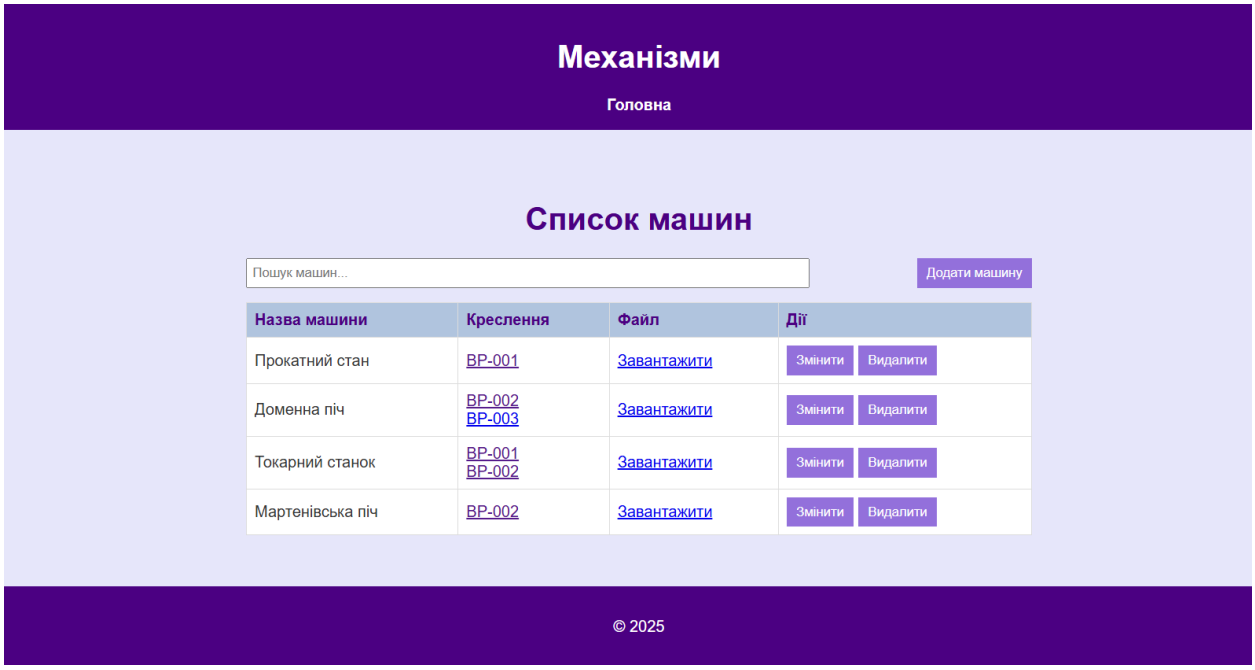


Рисунок *13 — Інтерфейс головної сторінки*

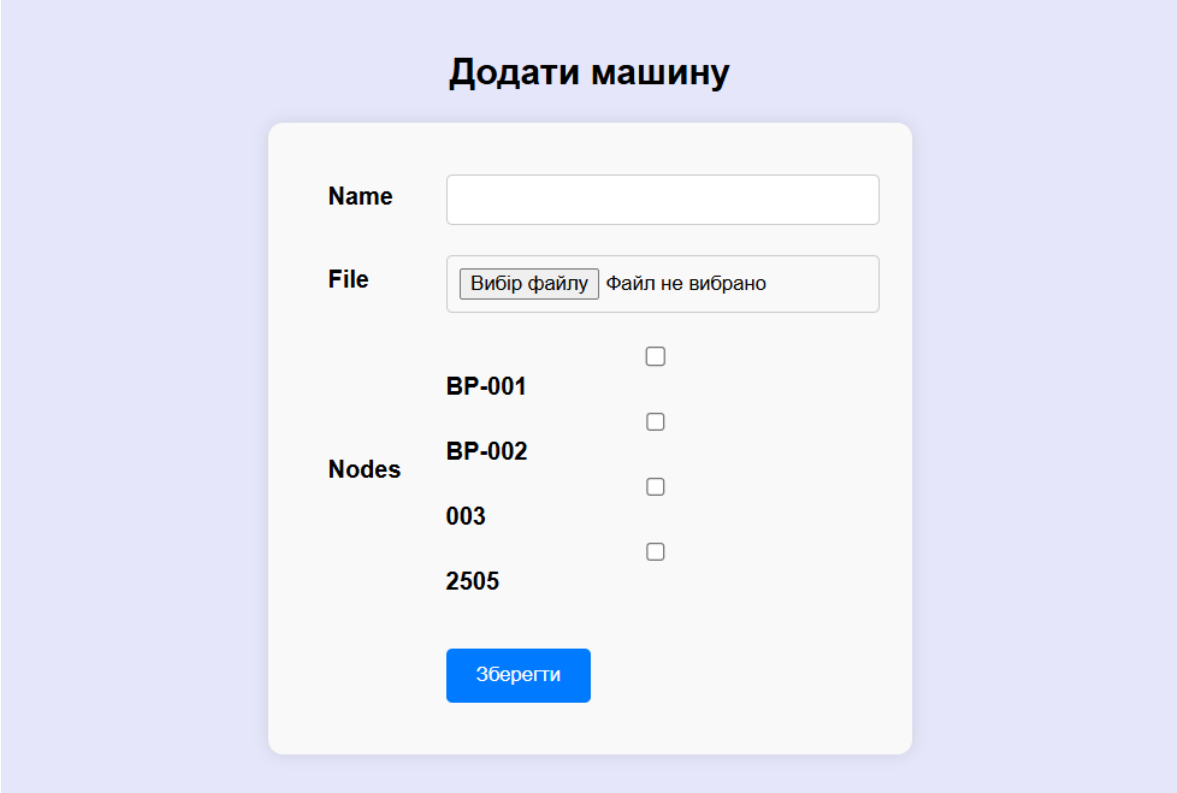


Рисунок *14 — Створення машини*

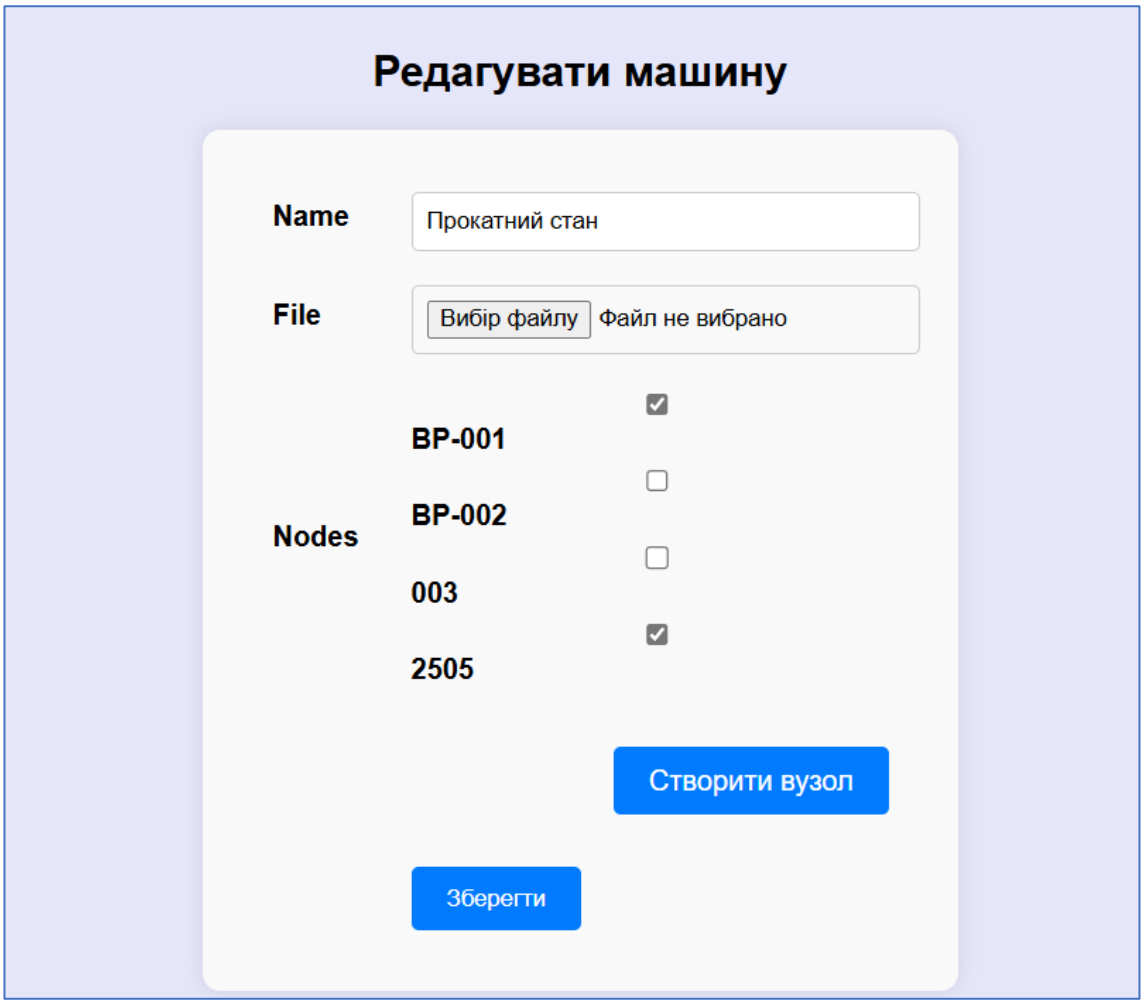


Рисунок *15 — Зміна даних машини*

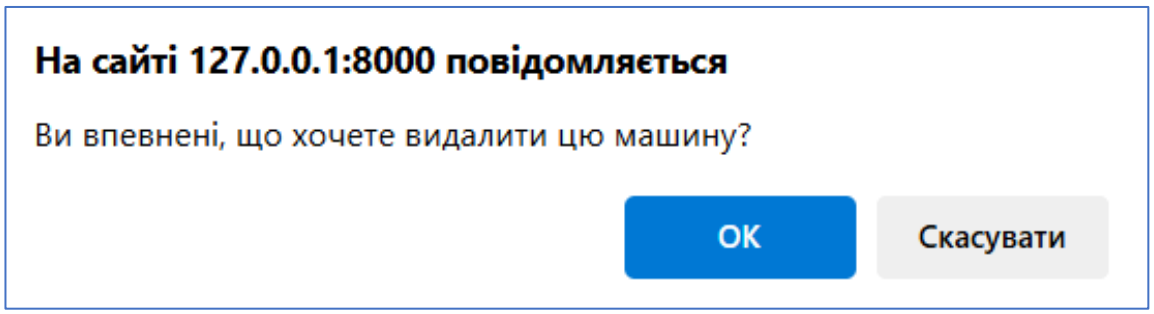


Рисунок *16 — Видалення машини*

При натисканні на вузол користувач перенаправляється на сторінку з даними про нього, що продемонстровано на рисунку 17. Де відображається його назва, мінімальна задана надійність, що використовується при розрахунках, та список елементів вузла, а саме: креслення, назва елемента, позиція по специфікації, кількість елементів даного типу у вузлі, ймовірність

відмови, ціна, необхідна кількість, файл, якщо такий завантажений. Можливість редагувати, додавати та видаляти елементи.

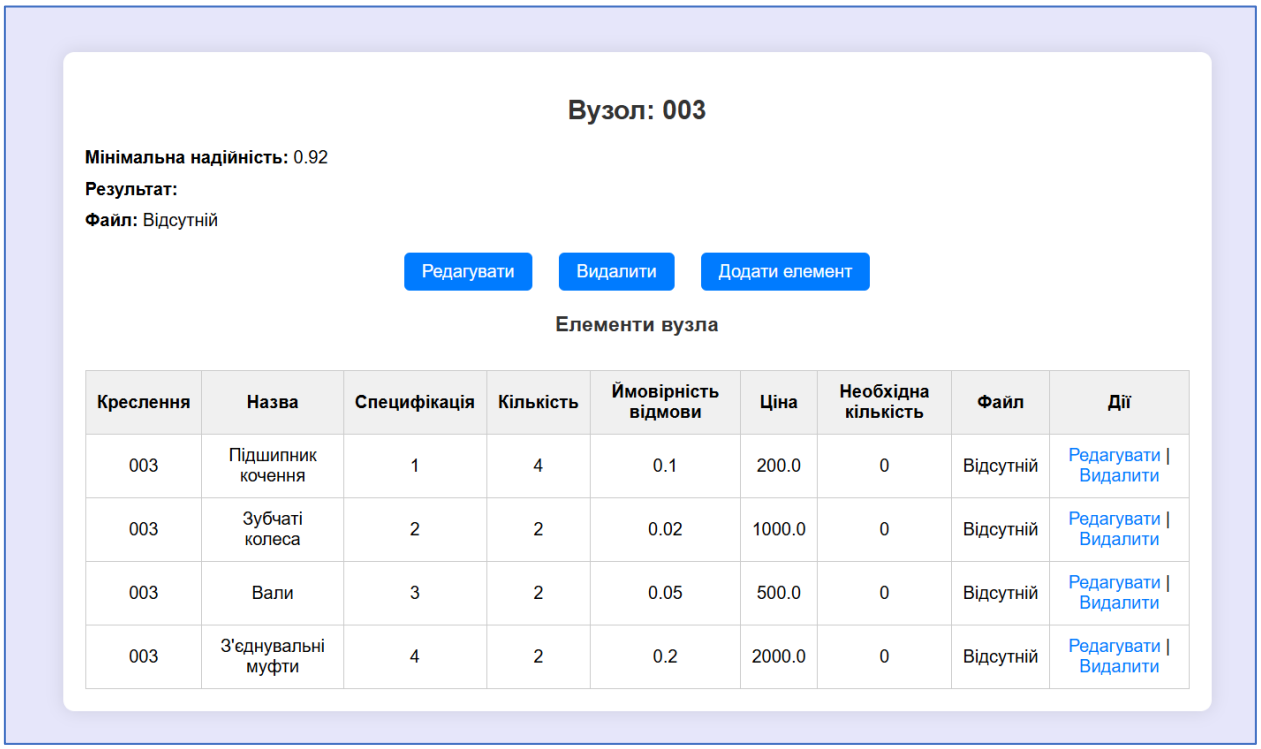


Рисунок *17 — Інформація про вузол*

Користувач вказує загальну кількість типів елементів, бажаний рівень надійності системи. Для кожного типу елемента вводяться кількість одиниць, вартість та ймовірність відмови. (рисунок 18), отримані дані зберігаються і використовуються для проведення розрахунку, для побудови домінуючого вектору прирощення достатності

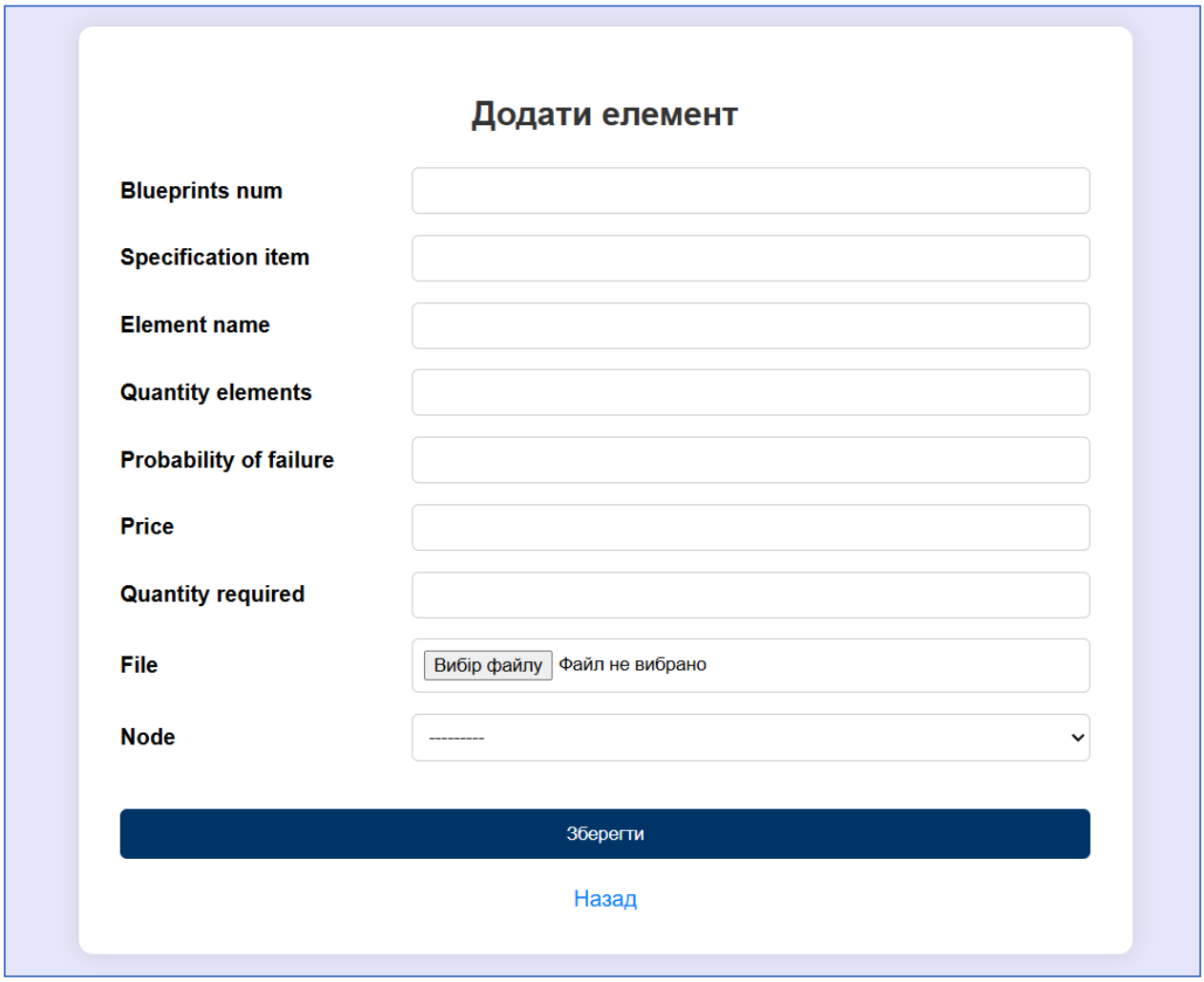


Рисунок *18 — Форма для додавання елементів*

Після обрання опції «Розрахувати» програма виконує обчислення на основі введених даних та обраних критеріїв. Результатом роботи є відображення оптимальної кількості кожного типу запасних частин, загальну вартість та отриманий рівень надійності системи. Також для зручності використання Алгоритм програми базується на принципах теорії надійності та методах оптимізації (рисунок 19).



Результати розрахунків

Вузол: 003

Загальна надійність системи: 0.95368

Назва елемента	Кількість	Ціна за одиницю	Загальна вартість (грн)	Кількість без урахування вартості
Підшипник кочення	6	200.0	1200.0	6
Зубчаті колеса	3	1000.0	3000.0	3
Вали	3	500.0	1500.0	3
З'єднувальні муфти	3	2000.0	6000.0	3
Сумарна вартість			**11700.0**	

Рисунок *19 — Результат розрахунків*

На основі розробленого додатку було отримано набір даних для специфікації редуктора привода моталки гарячої смуги для змотування рулонів з вісьмома формуючими роликами, кресленик якого представлений у додатку Г. Також у додатку В представлено загальний вид моталки гарячої смуги для змотування рулонів з вісьмома формуючими роликами.

Таблиця містить наступні характеристики: *Код деталі:* Унікальний код, який ідентифікує кожну запасну частину; *Назва деталі:* Назва або найменування деталі для легкого розуміння; *Кількість:* Зазначена кількість деталей згідно специфікації; *Вартість:* Вартість однієї одиниці деталі. *Ймовірність відмови:* Оцінка ймовірності того, що деталь вийде з ладу протягом певного періоду. *Потрібна кількість:* Кількість деталей, яку програма визначила як необхідну для забезпечення заданого рівня надійності системи. *Вартість (потрібна):* Загальна вартість необхідної кількості деталей. *Отримана надійність* (Лістинг 1)*:* Розрахований рівень надійності системи з урахуванням оптимального складу запасних частин. Сформований набір може бути використаний для широкого спектру аналітичних завдань і прогнозування.

Лістинг *1 — Моделі*

```
from django.db import models

class Node(models.Model):
    blueprints_num = models.CharField(max_length=100,
unique=True)
    min_reliability = models.FloatField()
    result = models.TextField(blank=True, null=True)
    file = models.FileField(upload_to='nodes_files/',
blank=True, null=True)

    def __str__(self):
        return self.blueprints_num

class Element(models.Model):
    blueprints_num = models.CharField(max_length=100)
    specification_item = models.CharField(max_length=100)
    element_name = models.CharField(max_length=100)
    quantity_elements = models.IntegerField()
    probability_of_failure = models.FloatField()
    price = models.FloatField()
    quantity_required = models.IntegerField(blank=True,
null=True)
    file = models.FileField(upload_to='elements_files/',
blank=True, null=True)
    node = models.ForeignKey(Node,
on_delete=models.CASCADE, related_name='elements',
null=True) #Зв'язок з таблицею Node

    def __str__(self):
        return self.element_name

class Machine(models.Model):
    name = models.CharField(max_length=200)
    file = models.FileField(upload_to='machines_files/',
blank=True, null=True)
    nodes = models.ManyToManyField(Node,
related_name='machines')
    def __str__(self):
        return self.name
```

Розрахункову частину додатку для обчислення потреби у запасних частинах за допомогою домінуючих векторів отриманого приросту достатності наведено у лістингу 2.

Лістинг *2 — Розрахункова частина домінуючих векторів*

```
from .models import Node
import math


def load_data_for_calculations(blueprints_num):
    try:
        node =
Node.objects.get(blueprints_num=blueprints_num)
        elements = node.elements.all()

        if not elements:
            print(f"Помилка: Вузол {blueprints_num} не має
елементів")
            return [], [], [], 0

        n1_values = [element.quantity_elements for element
in elements]
        c_values = [element.price for element in elements]
        q_values = [element.probability_of_failure for
element in elements]
        v = node.min_reliability

        print(f"Дані для розрахунку ({blueprints_num}):")
        print(f"n1_values: {n1_values}")
        print(f"c_values: {c_values}")
        print(f"q_values: {q_values}")
        print(f"Мінімальна надійність: {v}")

        print(f"Дані для розрахунку: n1={n1_values},
c={c_values}, q={q_values}, v={v}")

        return n1_values, c_values, q_values, v
    except Node.DoesNotExist:
        raise ValueError(f"Вузол {blueprints_num} не
знайдено.")

def main_from_db(blueprints_num):
    try:
        # Завантаження даних із бази
        n1_values, c_values, q_values, v =
load_data_for_calculations(blueprints_num)
```

```
        # Ініціалізація об'єктів для розрахунків
        n = len(n1_values)
        z_ch = [Zch() for _ in range(n)]
        for i in range(n):
            z_ch[i].n1 = n1_values[i]
            z_ch[i].c = c_values[i]
            z_ch[i].q = q_values[i]

        # Виконання розрахунків
        z_ch, p = step(z_ch, n, v)

        # Повернення результатів
        results = {'reliability': p, 'details': [(i + 1,
z.m) for i, z in enumerate(z_ch)]}
        return results
    except ValueError as e:
        return str(e)


class Zch:
    def __init__(self):
        self.n1 = 0  # Кількість деталей
        self.c = 0.0  # Вартість
        self.q = 0.0  # Ймовірність відмови
        self.m = 0  # Запас
        self.p1 = 0.0
        self.p2 = 0.0
        self.etta = 0.0


def pow(a, b):
    return math.pow(a, b)


def cc(a, b):
    if b > a:
        return 0
    S = 1.0
    for i in range(b + 1, a + 1):
        S *= i / (i - b)
    return S


def step(z_ch, n, v):
    for i in range(n):
        pq = 1.0
        osq = pow(1 - z_ch[i].q, z_ch[i].n1)
```

```
            for mi in range(z_ch[i].m + 1):
                z_ch[i].p1 += cc(z_ch[i].n1, mi) * pq * osq
                pq *= z_ch[i].q
                osq /= (1 - z_ch[i].q)

        p = 0
        while p < v:
            for i in range(n):
                osq = pow(1 - z_ch[i].q, z_ch[i].n1)
                p2 = 0.0
                pq = 1.0

                for mi in range(z_ch[i].m + 2):
                    p2 += cc(z_ch[i].n1, mi) * pq * osq
                    pq *= z_ch[i].q
                    osq /= (1 - z_ch[i].q)

                z_ch[i].etta = (p2 / z_ch[i].p1 - 1) /
z_ch[i].c

            m_e_i = max(range(n), key=lambda i: z_ch[i].etta)
            z_ch[m_e_i].m += 1

            p = 1.0
            for i in range(n):
                p *= z_ch[i].p1 if i != m_e_i else p2

        return z_ch, p
```

3.3 Тестування програмного продукту

При перевірці поведінки системи, було складено таблицю 3, де описано завдання, передумову, кроки, очікуваний результат та отриманий результат.

Таблиця *3 — Тест кейси*

№	Завдання	Передумова	Кроки	Очікуваний результат	Отриманий результат
1	Перевірити чи може користувач автроризуватися	Користувач зареєстрований	1. Зайти за посиланням “/login” 2. Ввести дані (Існуючий логін та пароль) 3. Натиснути «Увійти»	Відбуваєтьс я перенаправл ення на головну сторінку	Відбувається перенаправле ння на головну сторінку
2	Перевірити можливість додавати машини	Користувач авторизова ний	1. На головній сторінці натиснути «Додати машину» 2. Ввести дані (Назву, завантажити файл з кресленником, Обрати потрібні вузли) 3. Натиснути «Зберегти»	На головній сторінці відображає ться додана машина з відповідним и даними	На головній сторінці відображаєть ся додана машина з відповідними даними
3	Перевірити можливість редагуванн я машини	Користувач авторизова ний, машина додана	1. На головній сторінці натиснути «Змінити» навпроти потрібної машини 2. Ввести дані (Назву, завантажити файл з кресленником, Обрати потрібні вузли) 3. Натиснути «Зберегти»	На головній сторінці відображає ться відповідна машина зі зміненими даними	На головній сторінці відображаєть ся відповідна машина зі зміненими даними

Продовження таблиці *3*

№	Завдання	Передумова	Кроки	Очікуваний результат	Отриманий результат
4	Перевірити можливість видалення машин	Користувач авторизований, машина додана	1. На головній сторінці натиснути «Змінити» навпроти потрібної машини 2 Підтвердити видалення натиснувши кнопку «Ок» на спливаючому повідомлені	На головній сторінці не відображається відповідна машина	На головній сторінці не відображається відповідна машина
5	Перевірити можливість створити вузол	Користувач авторизований, машина додана	1. Навпроти відповідної машини натиснути «Змінити» 2. Натиснути «Додати вузол» 3. Ввести дані (Назва, надійність, файл) 4. Натиснути «Зберегти»	Додано новий вузол до списку, цей вузол можна додати до машини	Додано новий вузол до списку, цей вузол можна додати до машини

6	Перевірити можливість додати елемент	Створений вузол	1.Натиснути на вузол до якого потрібно додати елемент 2. Натиснути на “Додати елемнт” 3. Ввести дані (Назву, позицію по специфікації, кількість, вартість, ймовірність відмови, номер кресленика) 4. Натиснути кнопку «Зберегти»	У списку елементів вузла відображається доданий елемент	У списку елементів вузла відображається доданий елемент

Продовження таблиці *3*

№	Завдання	Передумова	Кроки	Очікуваний результат	Отриманий результат
7	Перевірити можливість редагувати елемент	Створений елемент	1.Натиснути на вузол де потрібно редагувати елемент 2. Натиснути на елемнт 3. Редагувати дані (Назву, позицію по специфікації, кількість, вартість, ймовірність відмови, номер кресленика) 4. Натиснути кнопку «Зберегти»	У списку елементів вузла відображаються редаговані дані про елемент	У списку елементів вузла відображаються редаговані дані про елемент
8	Перевірити розрахунок	Створений вузол	1. Зайти за посиланням «/calculate/BP-002/» де BP-002 є назвою вузла	Відображається таблиця з результатами та	Відображається таблиця з результатами та отримана надійність

				отримана надійність	

Необхідність мати великі обсяги історичних даних для побудови точних моделей прогнозування є серйозною перешкодою для ефективного управління запасами. Постійні зміни попиту, вихід нових продуктів та інші фактори ускладнюють збір достатньої кількості даних для традиційних методів прогнозування. Це особливо актуально для нових або малопопулярних деталей, де історичні дані можуть бути відсутні або недостатньо репрезентативними. В результаті, підприємства часто стикаються з труднощами при плануванні закупок і оптимізації рівнів запасів, що може призвести до дефіциту або надлишків товарів.

За допомогою розробленого додатку було згенеровано детальний звіт про оптимальний склад запасних частин для моталки гарячої смуги для змотування рулонів з вісьмома формуючими роликами з урахуванням поточної вартості деталей, з передбаченою інфляцією вартості а також для варіанту без урахування вартості (Таблиця 4).

Таблиця *4 — Результати розрахунків на прикладі специфікації*

Позиція по специфікації	Назва елемента	Кількість	Вартість	Ймовірність відмови	Потрібна кількість при розрахунку з урахуванням вартості	Вартість при розрахунку з урахуванням вартості	Потрібна кількість при розрахунку без урахування вартості	Вартість при розрахунку без урахування вартості
1	2	3	4	5	6	7	8	9
2	Корпус редуктора	1	171500	0,03	0	0	0	0
3	Кришка редуктора	1	86452	0,03	0	0	0	0
4	Віддушина	1	478	0,01	1	478	0	0
5	Кільце	1	94	0,005	1	94	0	0
6	Прокладка	1	568	0,07	1	568	1	568
7	Кришка	1	468	0,005	0	0	0	0
8	Кільце	1	89	0,005	1	89	0	0
9	Кільце масловідбиваюче	1	150	0,02	1	150	0	0
10	Штифт конічний 30Х120	2	270	0,07	2	540	1	270
11	Підшипник 7526	2	1856	0,18	2	3712	1	1856
12	Кришка d=68 6 болтів	1	320	0,005	1	320	0	0
13	Вал-шестерня	1	12050	0,12	0	0	1	12050
14	Прокладка набір 6 отворів	1	786	0,07	1	786	1	786
15	Кришка	2	812	0,005	1	812	0	0
16	Пружина НР-330	1	615	0,06	1	615	1	615
17	Пружина 330	1	612	0,06	1	612	1	612
18	Болт чистий М20х50	6	40	0,005	1	40	0	0

Продовження таблиці *4*

1	2	3	4	5	6	7	8	9
19	Болт чистий М6х16	14	12	0,005	2	24	1	12
20	Ущільнення НР-120	1	2836	0,07	1	2836	1	2836
21	Пружина 120	1	348	0,06	1	348	1	348
22	Болт чистий М16х50	6	17	0,005	2	34	0	0
23	Вантажопідйомне кільце м2У	2	120	0,04	2	240	1	120
24	Болт чистий М10х25	2	9	0,005	1	9	0	0
25	Болт чистий М30х110	8	532	0,005	1	532	1	532
26	Гайка А М30	8	69	0,005	1	69	1	69
27	Шайба пружинна полегшена	8	75	0,005	1	75	1	75
28	Шпонка призм. скругл. М32х18х150	1	175	0,23	4	700	2	350
29	Болт напівчистовий М48х490	2	6880	0,005	0	0	0	0
30	Гайка А М48	4	221	0,07	2	442	1	221
31	Пробка М-30	1	1680	0,005	0	0	0	0
32	Шайба ущільнювальна ø33х45	1	165	0,07	2	330	1	165
33	Масловказівник	1	249	0,005	1	249	0	0
34	Зубчасте колесо	1	16350	0,17	1	16350	1	16350
35	Роликопідшипник 22 37 860	1	9374	0,28	2	18748	3	28122

На рисунках 20, 21 представлено порівняння рівнів надійності роботи обладнання та загальної вартості необхідної кількості деталей за різними сценаріями розрахунків.

Порівняння результатів моделей з урахуванням та без урахування вартості дозволяє оцінити вплив економічних факторів на оптимальний склад запасних частин та вибрати найбільш вигідний варіант для підприємства.

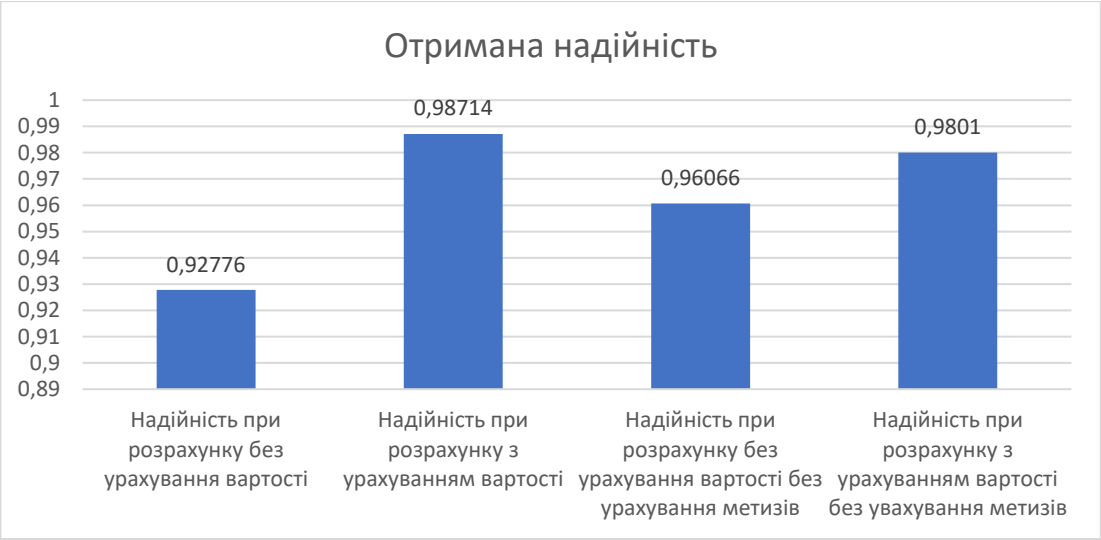


Рисунок *20 — Порівняння рівнів надійності системи за різними сценаріями розрахунку*



Рисунок *21 — Порівняння загальної вартості необхідної кількості деталей за різними сценаріями розрахунків*

ВИСНОВКИ

Проведене дослідження та розроблена програмна система мають особливу актуальність для машинобудівних підприємств Запорізького регіону. Регіон має значний потенціал у сфері машинобудування, і оптимізація управління запасами є критично важливою для підвищення ефективності виробництва та конкурентоспроможності підприємств.

Запропонований підхід, що поєднує класичний метод домінуючих послідовностей з індивідуальними характеристиками обладнання, дозволяє точніше прогнозувати потребу в запасних частинах та мінімізувати пов'язані з цим витрати. Результати дослідження підтверджують ефективність розробленого програмного забезпечення та його потенціал для широкого застосування в промисловості.

Розроблена програмна система дозволяє автоматизувати рутинні операції, пов'язані з управлінням запасами, зменшити ризик дефіциту або надлишку запасних частин, а також оптимізувати витрати на їх зберігання.

Крім того, розроблена система може бути ефективним інструментом для навчання студентів технічних спеціальностей. За допомогою цієї системи студенти зможуть наочно ознайомитися з принципами роботи систем управління запасами, отримати практичні навички з використання сучасних програмних інструментів для аналізу даних та прийняття рішень, розробити власні моделі прогнозування та оптимізації, використовуючи систему як платформу для експериментів, глибоко зрозуміти взаємозв'язки між різними параметрами системи управління запасами та їх вплив на ефективність виробництва.

Таким чином, розроблена система може стати важливим інструментом для розвитку промислового потенціалу Запорізького регіону та зміцнення його позицій на внутрішньому та зовнішньому ринках.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ukrainian W. Уроки від W3Schools українською онлайн. W3Schools українською. Безплатні уроки онлайн для початківців, школярів та студентів. URL: https://w3schoolsua.github.io/django/index.html#gsc.tab=0 (дата звернення: 01.04.2025).

2. Vincent W. S. Django for professionals: production websites with python & django. Independently published, 2019. 370 p.

3. Василевський, О. М., Поджаренко В. О. Нормування показників надійності технічних засобів. Вінниця : ВНТУ, 2010. 129 с.

4. Таратута К. В., Шанько О. Ю. Вибір та обґрунтування номенклатури показників теоретичної надійності металургійного обладнання. *Збірник наукових праць "Металургія"*. 2015. № 2(34). С. 109-113. URL: https://oldzdia.znu.edu.ua/gazeta/EnglishMetallurgy_34_22.pdf

5. Канарчук В.С., Полянський С.К., Дмітрієв М.М. Надійність машин: підручник. Київ: Либідь, 2003. 424 с.

6. . Кравченко В. Технічна діагностика. Контроль та прогнозування. Монографія / А.Я.Жук, Г.П. Малишев, Н. К. Желябіна, О.М. Клевцов; ЗДІА. Запоріжжя : ЗДІА, 2008. 499 с.

7. Седуш В.Я. Надійність, ремонт і монтаж металургійних машин : підручник. Донецьк: ТОВ «Юго – Восток, Лтд», 2008. 379 с.

8. Srivastava A. K. Django , The Python Web Framework. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*. 2022. Vol. 06, no. 05. URL: https://doi.org/10.55041/ijsrem13183 (date of access: 11.06.2025).

9. Martin R. C. Clean Craftsmanship: Disciplines, Standards, and Ethics. Pearson Education, Limited, 2021.

10. Murach's MySQL (4th Edition). Mike Murach & Associates, Inc., 2023.

11. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Pearson Education, Limited, 2018.

12. REGULARITIES AND FEATURES OF NEED FORMATION IN SPARE PARTS AT AUTO SERVICE ENTERPRISES / S. O та ін. *The National Transport University Bulletin*. 2022. Т. 3, № 53. URL: https://doi.org/10.33744/2308-6645-2022-3-53-329-336 (дата звернення: 19.03.2025).

13. Ремонт металургійного обладнання / А. Я. Жук та ін. Київ : Кондор, 2017. 233 с.

14. Монтаж металургійного обладнання / А. Я. Жук та ін. Київ : Кондор, 2018. 329 с.

15. Власова Л. А., Міхайлуца О. М. Стратегії та методи розрахунку потреби в запасних частинах. *Матеріали* : XIV Всеукр. конф. молодих вчен. «МОЛОДІ ВЧЕНІ 2024 − ВІД ТЕОРІЇ ДО ПРАКТИКИ», м. Дніпро, 21 берез. 2024 р. Дніпро, 2025. С. 132–134.

16. Власова Л. А., Міхайлуца О. М. Програмне забезпечення для розрахунку потреби у запасних частинах. *ЗБІРНИК наукових праць студентів, здобувачів PhD, докторантів і молодих вчених «МОЛОДА НАУКА-2025» Том V «Секції Інженерного навчально-наукового інституту ім. Ю.М. Потебні Запорізького національного університету»* : XVIII унів. науково-практ. конф. студентів, аспірантів, докторантів і молодих вчен. «Молода наука-2025»,, м. Запоріжжя, 14–18 квіт. 2025 р. Запоріжжя, 2025. С. 382–384.

17. Власова Л. А., Міхайлуца О. М. *Збірник доповідей* : Міжнар. науково-практ. інтернет-конф. Молодь в науці: дослідж., проблеми, перспективи, м. Вінниця, 15 листоп. 2023 р. – 20 черв. 2024 р. Вінниця, 2023. С. 338–339.

18. Власова Л. А., Міхайлуца О. М. Автоматизація розрахунків запасних частин для капітального ремонту: опис, можливості та перспективи розвитку. *ІНЖЕНЕРНІ ІННОВАЦІЇ ТА РОЗБУДОВА НАЦІОНАЛЬНОЇ ЕКОНОМІКИ матеріали І міжнародної науково-практичної конференції інженерного навчально-наукового інституту ім. Ю. М. потебні* : І Міжнар. науково-практ. конф. Інж. навчально наук. ін-ту ім. Ю. М. Потебні Запоріз. нац.

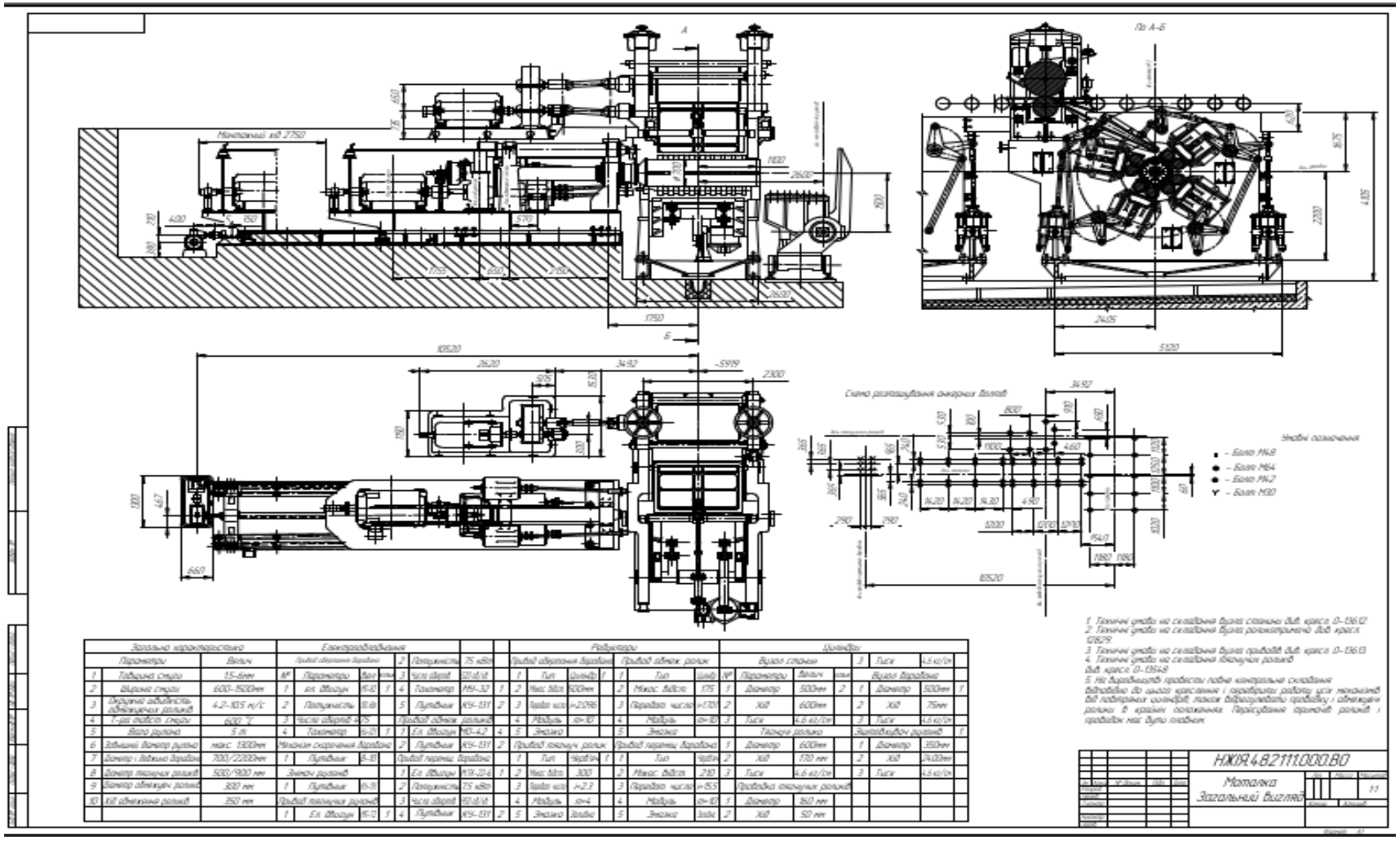
ун-ту «Інж. інновації та розбудова нац. економіки», м. Запоріжжя, 9–10 трав. 2024 р. Одеса, 2024. С. 836–838.

19. Довідник з HTML URL: https://css.in.ua/html/tags/ (дата звернення 22.04.2025)

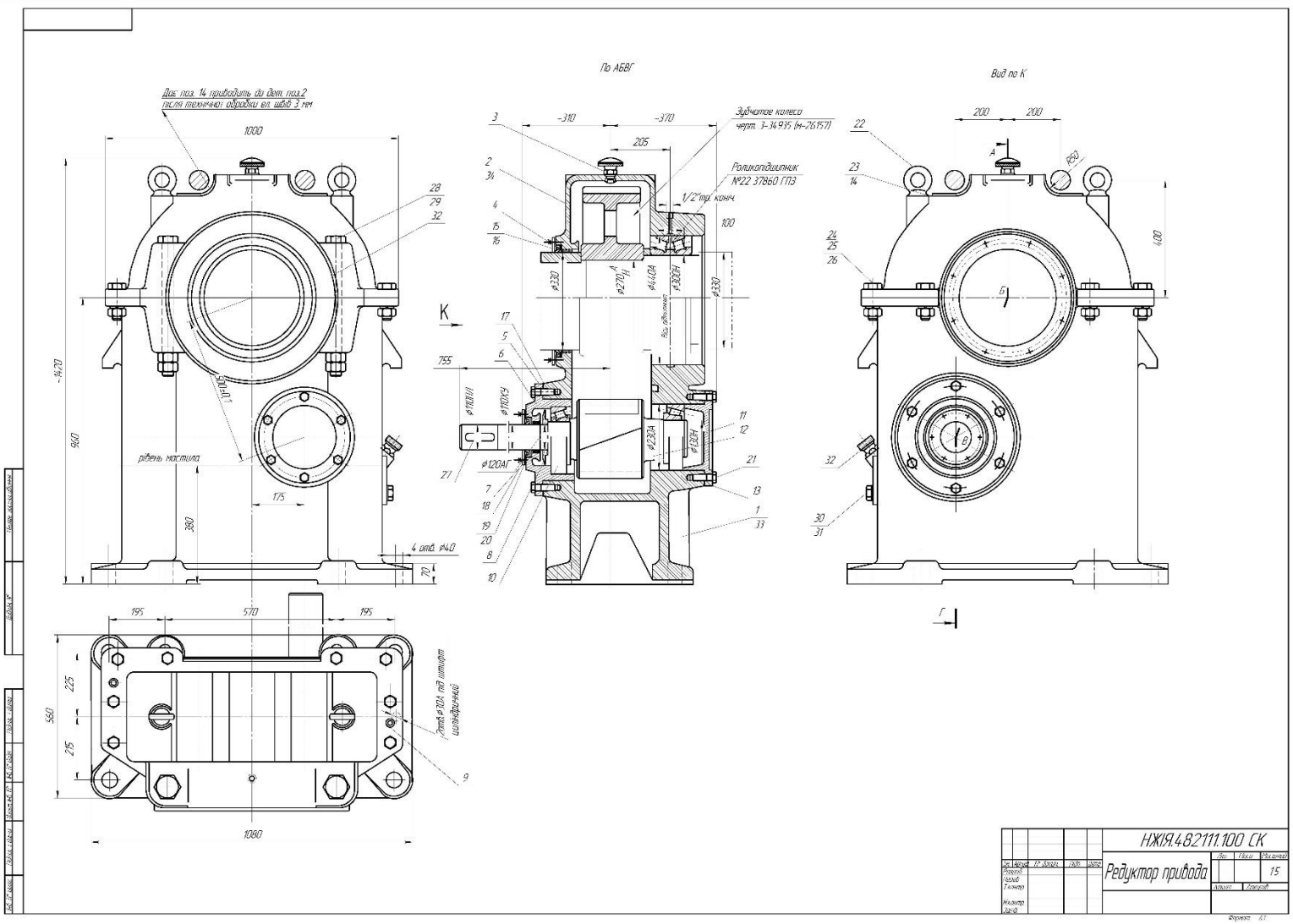
20. Довідник CSS URL : https://css.in.ua/css/properties/ (дата звернення 22.04.2025)

21. Pycharm це IDE від JetBrains для програмування на Python. *FoxmindEd*. URL: https://foxminded.ua/pycharm-tse/ (дата звернення: 22.04.2025).

ДОДАТОК А КРЕСЛЕНИК МОТАЛКИ ГАРЯЧОЇ ШТАБИ ДЛЯ ЗМОТУВАННЯ РУЛОНІВ З ВІСЬМОМА ФОРМУЮЧИМИ РОЛИКАМИ



ДОДАТОК Б КРЕСЛЕНИК РЕДУКТОРА ПРИВОДА МОТАЛКИ ГАРЯЧОЇ СМУГИ ДЛЯ ЗМОТУВАННЯ РУЛОНІВ З ВІСЬМОМА ФОРМУЮЧИМИ РОЛИКАМИ



Акт про впровадження

УКРАЇНА
Міністерство освіти і науки України
Запорізький національний університет
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-
НАУКОВИЙ ІНСТИТУТ
ім. Ю.П. Потебні
пр. Соборний, 226, м. Запоріжжя, 69006
телефон/факс (061) 227-12-72
e-mail: ei@znu.edu.ua
18.09.2024 № 319/41.01-31
На № _____ від _____

А К Т

про впровадження результатів роботи проєкту розробки програмного забезпечення в освітній процес

Цей акт складений у тому, що в навчальному процесі кафедри металургійного обладнання Інженерного навчально-наукового інституту ім. Ю.М. Потебні ЗНУ використовуються результати роботи проєкту «Програмне забезпечення для розрахунку потреби в запасних частинах», виконаного студенткою кафедри електроніки, інформаційних систем та програмного забезпечення Власовою Лілією Андріївною, під керівництвом к.т.н., доцента, доцента кафедри електроніки, інформаційних систем та програмного забезпечення Міхайлуци Олени Миколаївни. Програмне забезпечення дозволяє розрахувати необхідну кількість запасних частин, при підготовці до проведення капітального ремонту, враховуючи або не враховуючи вартість елементів за потребою користувача, на підставі ймовірностей безвідмовної роботи вузлів та деталей.

Результати отримані в роботі використовуються при викладанні дисциплін «Монтаж, експлуатація та ремонт металургійного обладнання» та «Надійність металургічних систем».

Директор Інженерного
навчально-наукового
Інститут ім. Ю.М. Потебні ЗНУ Наталя МЕТЕЛЕНКО

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ім. Ю.М. Потебні
м. Запоріжжя Україна

Декларація

академічної доброчесності

здобувача ступеня вищої освіти ЗНУ

Я, Власова Лілія Андріївна студентка 4 курсу, форми навчання денної, Інженерного навчально-наукового інституту ім. Ю.М. Потебні ЗНУ, спеціальність 121 Інженерія програмного забезпечення освітньої програми “Програмне забезпечення систем”, адреса електронної пошти se121-28@stu.zsea.edu.ua, — підтверджую, що написана мною кваліфікаційна робота на тему «**Програмне забезпечення для розрахунку потреби у запасних частинах»** відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлена.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;
- згодна на перевірку моєї роботи на відповідність критеріям академічної добро чесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 10.06.2025 _____ Власова Л. А.
(підпис) (прізвище та ініціали) (Студент)

Дата 10.06.2025 _____ Міхайлуца О. М.
(підпис) (прізвище та ініціали) (Керівник)