

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут

комп'ютерних наук та управління проектами

(повне найменування інституту, назва факультету)

Програмного забезпечення автоматизованих систем

(повна назва кафедри)

Пояснювальна записка до кваліфікаційної (магістерської) роботи

за темою **Розробка моделі верифікації інформаційно-вимірювальної
системи з обмеженими ресурсами**

Виконав: студент 6 курсу, групи 6151м

спеціальності 152 «Метрологія та
інформаційно-вимірювальна техніка»

_____ Михайлюк М. В.

_____ (підпис, прізвище та ініціали)
Керівник Пономаренко Т.В.

(підпис, прізвище та ініціали)

Рецензент _____ Приходько С.Б.

(підпис, прізвище та ініціали)

Завідувач кафедри _____ Приходько С.Б.

(підпис, прізвище та ініціали)

м. Миколаїв – 2020 р.

Міністерство освіти і науки України
Національний університет кораблебудування
імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем

Освітній ступень Магістр

Галузь 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри

“ 26 ” 10 2020 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ (МАГІСТЕРСЬКУ) РОБОТУ СТУДЕНТУ
Михайлюку Михайлу Володимировичу

1. Тема магістерської роботи _____ Розробка моделі верифікації інформаційно-вимірювальної системи з обмеженими ресурсами

керівник роботи Пономаренко Тетяна Вікторівна, к.т.н., доцент

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

затверджені наказом вищого навчального закладу від “ 26 ” жовтня 2020 року №1037-
уч _____

2. Строк подання студентом роботи 01.12.2020 року

3. Вихідні дані до роботи __завдання__ дипломного керівника

4. Зміст магістерської роботи (МР):

- Титульний аркуш, завдання на кваліфікаційну (магістерську) роботу, реферат (українською, англійською), зміст, перелік умовних позначень, символів, одиниць та термінів (за необхідності).

- Вступ (Актуальність теми. Зв'язок роботи з науковими програмами, планами, темами. Мета і завдання дослідження. Об'єкт дослідження. Предмет дослідження. Методи дослідження. Наукова новизна одержаних результатів. Практичне значення одержаних результатів. Особистий внесок здобувача. Апробація результатів досліджень. Публікації.)

- Огляд літератури за темою, обґрунтування необхідності проведення досліджень за обраною темою, вибір напрямків досліджень, мета дослідження, основні задачі дослідження

- Викладення результатів власних досліджень з висвітленням того нового, що пропонується

- Проект програмного забезпечення

- Результати досліджень та розробки проекту програмного забезпечення

- Організаційно-економічний розділ _____

- Розділи з охорони праці та охорони навколишнього середовища _____

- Висновки

- Список використаних джерел

- Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма і методика випробувань програмного забезпечення)

РЕФЕРАТ

Михайлюк Михайло Володимирович

«Розробка моделі верифікації інформаційно-вимірювальної системи з обмеженими ресурсами»

Кваліфікаційна робота на здобуття освітнього рівня магістра зі спеціальності 152 «Метрологія та інформаційно-вимірювальна техніка». Національний університет кораблебудування імені адмірала Макарова. Миколаїв, 2020 р.

Обсяг роботи: 91 стор., 22 табл., 21 рис., 17 використаних джерел, 5 додатків.

Актуальність теми роботи: необхідність створення моделі верифікації для різних конфігурацій апаратного забезпечення з обмеженими ресурсами.

Мета та завдання дослідження. Метою даної магістерської роботи є підвищення надійності функціонування систем з обмеженими ресурсами з реконфігурованих компонент за рахунок забезпечення цілісності даних і автоматизації процесу верифікації.

Завдання дослідження: проаналізувати існуючі моделі верифікації, інформаційно-вимірювальних систем з обмеженими ресурсами та створити модель верифікації, що описує адаптивну архітектуру вбудованих систем і містить період передачі результатів верифікації під зовнішнє сховище, що дозволяє застосовувати дану модель для різних конфігурацій апаратного забезпечення.

Об'єктом дослідження інформаційно-вимірювальна система з обмеженими ресурсами.

Предмет дослідження - модель верифікації інформаційно-вимірювальної системи з обмеженими ресурсами.

Методи дослідження. У вирішенні поставлених завдань використано методи статистичного аналізу.

Наукова новизна одержаних результатів. Розроблено модель верифікації інформаційно-вимірювальної системи з обмеженими ресурсами, яка, на відміну від моделі верифікації веб-орієнтованих систем, описує адаптивну архітектуру вбудованих систем і містить період передачі результатів верифікації під зовнішнє сховище, що дозволяє застосовувати дану модель для різних конфігурацій апаратного забезпечення та підвищити надійність функціонування систем з обмеженими ресурсами з реконфігурованих компонент за рахунок забезпечення цілісності даних і автоматизації процесу верифікації.

Практичне значення одержаних результатів. Розроблено програму для верифікації інформаційно-вимірювальної системи.

Публікації. Результати роботи опубліковано у матеріалах Всеукраїнської науково-практичної інтернет-конференції матеріали II Всеукр. наук.-практ. конф. здобувачів вищої освіти й молодих учених, м. Кропивницький, 25–27 листоп. 2020 р.

Ключові слова: МОДЕЛЬ ВЕРИФІКАЦІЇ, ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНА СИСТЕМА, СИСТЕМИ З ОБМЕЖЕНИМИ РЕСУРСАМИ, : GRID, ETHERNET, ЦИКЛИЧЕСКОЕ ПЛАНИРОВАНИЕ, ПАКЕТ ЗАДАНИЙ, RASPBERRY PI.

ABSTRACT

Myhajluk Myhajlo

«Development of a model for verification of the information-measuring system
with limited resources»

The qualification work in obtaining a master's degree in specialty 152 - "152
"Metrology and information-measuring equipment". Admiral Makarov National
University of Shipbuilding. Mykolayiv, 2020.

The qualification work is presented on the 91 pages of typewritten text, contains
22 tables, 21 figures, 5 appendices and 17 references.

Relevance of the topic of the work: the need to create a verification model for
different configurations of hardware with limited resources.

The goal and objectives of the study. The goal is to create a verification
model that describes the adaptive architecture of embedded systems and allows you
to apply this model to different hardware configurations.

The objectives of the study: to analyze existing verification models,
information and measurement systems with limited resources and create a
verification model that describes the adaptive architecture of embedded systems and
contains the period of transfer of verification results to external storage, which allows
to apply this model to different hardware configurations.

The object of the study is information and measurement systems.

The subject of the study is a model of verification of information-measuring
system with limited resources.

The methods of the study. Methods of probability theory and mathematical
statistics were used in solving the tasks.

The scientific novelty of obtained results. A model of verification of information-measuring system with limited resources has been developed, which, unlike the model of verification of web-oriented systems, describes the adaptive architecture of embedded systems and contains the period of transfer of verification results to external storage, which allows applying this model to different hardware configurations.

The practical significance of obtained results Developed a program for verification of information and measurement system

Approbation of study results. The research results were published at the All-Ukrainian scientific-practical Internet conference

Publications. The results of the work were published in the materials of the All-Ukrainian scientific-practical internet-conference materials of the II All-Ukrainian. scientific-practical conf. applicants for higher education and young scientists, Kropyvnytskyi, November 25-27. 2020 / Ministry of Education and Science of Ukraine, Gos. Science. Institution "Institute for Modernization of Educational Content", Central Ukrainian nat. tech. un-t. - Kropyvnytskyi: CNTU, 2020. - P. 10.

Keywords: VERIFICATION MODEL, INFORMATION MEASUREMENT SYSTEM, SYSTEMS WITH LIMITED RESOURCES.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	8
ВСТУП.....	9
1 АНАЛІЗ ПРОЦЕСУ ВЕРИФІКАЦІЇ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНИХ СИСТЕМ З ОБМЕЖЕНИМИ РЕСУРСАМИ.....	11
2 РОЗРОБКА МОДЕЛІ ВЕРИФІКАЦІЇ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ З ОБМЕЖЕНИМИ РЕСУРСАМИ.....	15
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕРИФІКАЦІЇ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ З ОБМЕЖЕНИМИ РЕСУРСАМИ.....	19
4 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	27
5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	30
6 ОХОРОНА ПРАЦІ.....	35
7 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ.....	54
ДОДАТОК Б – ІНСТРУКЦІЯ КОРИСТУВАЧА.....	59
ДОДАТОК В – ТЕКСТ ПРОГРАМИ.....	65
ДОДАТОК Г – ТЕСТУВАННЯ	81
ДОДАТОК Д – ОПИС ПРОГРАМИ.....	88
ДОДАТОК Є – ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ.....	89

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Φ - функціональна одиниця;

c - концепція;

C - набір понять (композиційних і атомарних);

CA - набір атомарних концепцій;

DB - сховище даних;

e - підмножина ресурсів інформаційної системи;

E - набір ресурсів інформаційної системи;

ea - елементарна одиниця елементів інтерфейсу інформаційно-вимірювальної системи;

ed - елемент інформаційної системи;

Ed - набір web-ресурсів інформаційної системи;

Eh - набір адаптивних елементів апаратного забезпе- печення інформаційної системи;

f - функціональна одиниця системи;

F - безліч функціональних одиниць системи;

fa - адаптивна функція перетворення між еле ментами інформаційної системи і вимогами користувача;

I безліч вхідних даних;

interprototype - множина елементів прототипу інтерфейсу;

IoT - Internet of things;

MV - модель для верифікації.

ВСТУП

Актуальність теми. Розглянуто задачу розробки моделі верифікації інформаційно-вимірювальної системи з обмеженими ресурсами, де виміри енергетичних показників проводяться на реєстраторах, які надають інформацію щодо вимірів обсягів енергетичних показників за допомогою Raspberry Pi. Ключовим моментом в таких пристроях є їх обмеженість в системних ресурсах, але це не повинно ставити під удар усю глобальну інформаційно-вимірювальну систему, що призводить до посилення вимог до верифікації таких систем. Тому актуальність розробки моделі верифікації для таких систем для підтвердження їх стабільного функціонування є актуальним завданням.

Методи дослідження Поставлені завдання магістерської роботи вирішено з використанням методів обробки статистичних даних та математичного моделювання.

Наукова новизна одержаних результатів. Розроблено модель верифікації інформаційно-вимірювальної системи з обліку енергії з обмеженими ресурсами яка, за рахунок забезпечення цілісності даних і автоматизації процесу верифікації, дає підвищення надійності функціонування систем з обмеженими ресурсами з реконфігурованих компонент.

Практичне значення одержаних результатів. Розроблено програмне забезпечення для автоматизованої верифікації, що реалізує розроблену модель.

Особистий внесок здобувача Магістерська робота є самостійно виконаною працею. Усі результати, викладені у роботі, отримані автором особисто.

Апробація результатів стажування. Результати досліджень були оприлюднені на конференції “Комп’ютерна інженерія і кібербезпека : досягнення та інновації II Всеукр. наук.-практ. конф. здобувачів вищої освіти й молодих учених, м. Кропивницький, 25–27 листоп. 2020 р.

.

Публікації. За результатами досліджень було опубліковано 1 друковану роботу у співавторстві с керівником роботи у збірнику тез.

1 АНАЛІЗ ПРОЦЕСУ ВЕРИФІКАЦІЇ ІНФОРМАЦІЙНО-ВІМІРЮВАЛЬНИХ СИСТЕМ З ОБМЕЖЕНИМИ РЕСУРСАМИ

1.1 Огляд існуючих алгоритмів

Облік енергоресурсів має колосальне значення в їх виробництві та споживанні. В останні час в зв'язку з подорожчанням енергоресурсів і становленням оптового ринку електроенергії, тепла, води, газу і т.д. - виникла необхідність виділити в окрему галузь вимір і облік всіх видів енергоресурсів.

Подібні системи призначені для організації оперативного обліку споживання енергоресурсів та повинні мати наступні характеристики:

- Вимірювання кількості споживаної електроенергії, води, тепла, газу та інших енергоносіїв в поточний момент часу, за звітний період тощо.
- Контроль за технічними показниками споживання енергоресурсів (Наприклад: електроенергія, гаряча вода, газ, тепло з фіксацією кількості, обсягів, температур, тиску тощо).
- На основі моніторингу даних здійснювати довгострокове і короткострокове планування обсягів споживання енергоресурсів.
- Можливість обміну інформацією з головними і суміжними постачальниками енергоресурсів.
- Підготовку даних для фінансових звітів, систем білінгу, особистих кабінетів користувача.
- Можливість проведення активних заходів з енергозбереження як підприємствам, так і кінцевим користувачам.
- Повна зміна бізнес-процесів підприємств, якісно новий сервіс для споживачів енергоресурсів.

Для досягнення поставлених цілей, система обліку повинна володіти наступними основними якостями: достовірністю вимірювання; зручністю експлуатації; оперативністю інформації, що представляється; 4. потужними

аналітичними можливостями; розвиненими комунікаційними функціями; апаратною незалежністю; підтримкою міжнародних стандартів міжсистемних обмінів.

Система є проектно- і програмно-компонованим багатофункціональним засобом вимірювання і являє собою комплекс сертифікованих засобів обліку (вимірювань) енергоресурсів, програмного забезпечення і допоміжних засобів збору і передачі даних, що функціонально складається з:

- первинних засобів вимірювань - лічильників;
- пристроїв передачі інформації - логерів;
- пристроїв збору даних та виводу їх у зручному для користувачів вигляді -

Дашбоарду. (Рисунок 1.1)



Рисунок 1.1 - Склад інформаційно - вимірювальної системи

Первинні засоби вимірювання - це багатотарифні інтелектуальні пристрої, що мають інформаційні виходи. В якості первинних засобів вимірювання використовуються як правило лічильники електрики, води, тепла, сертифіковані в тій країні, на території якої здійснюється проект.

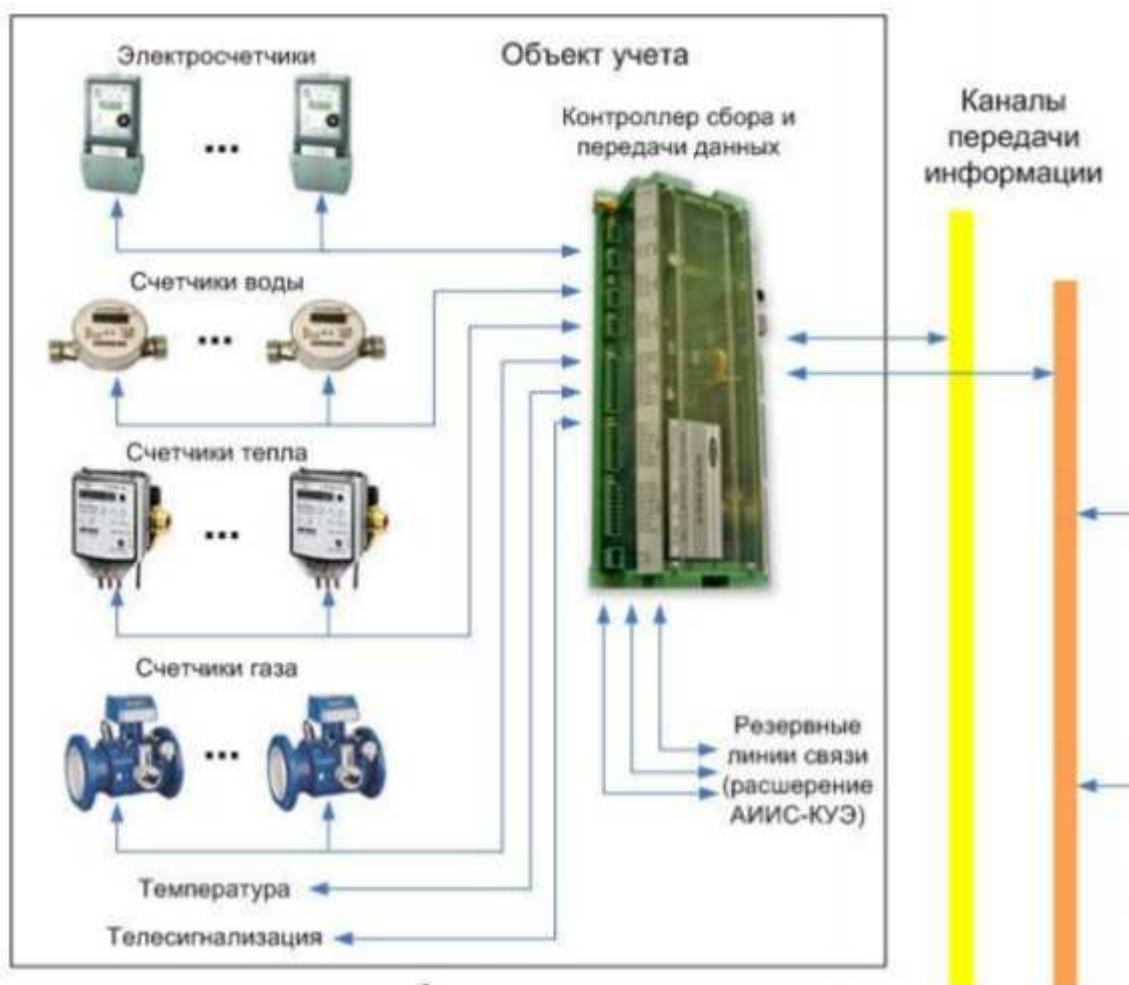


Рисунок 1.2 – Апаратні складові інформаційно-виміральної системи з обмеженими ресурсами

Верифікація систем з обмеженими ресурсами повинна забезпечувати достовірність і логічну цілісність даних, що необхідно для функціонування системи відповідно до стандартів якості [1-2]. Також, з огляду на обмеженість обсягу сховищ даних, результати верифікації необхідно передавати під зовнішнє сховище через певний період часу [3].

Метод верифікації моделей працює не безпосередньо з програмою, а з її моделлю. Щоб побудувати таку модель спочатку для системи створюється специфікація - набір властивостей, якому вона повинна задовольняти. Оскільки в методі model checking специфікація перевіряється на моделі, необхідно, щоб в моделі виявлялися все властивості, які присутні в специфікації. У той же час, не

має сенсу додавати в модель ті деталі системи, які не впливають на виконання необхідних властивостей [4].

Найбільший інтерес представляє завдання верифікації реактивних систем [5]. Такі системи не можна моделювати в термінах входу-виходу, так як їх робота може тривати нескінченно довго, і тому для них верифікується поведінка в часі. Для цього вводиться поняття стану системи, під яким в загальному випадку розуміється її миттєвий опис з фіксованими значеннями всіх змінних системи. Роботу реактивної системи при цьому можна представити у вигляді переходів між станами.

2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ВЕРИФІКАЦІЇ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ З ОБМЕЖЕНИМИ РЕСУРСАМИ

З системою, для якої буде створюватися модель, працюють користувачі контролери, які вносять до системи специфікацію логера - коректний валідаційний інтервал, критичний час.

Для визначення об'єкта тестування в моделі верифікації web-орієнтованих систем виділено елемент ed - ресурс, або документ, доступний через мережу і

$$rd : C_A \rightarrow E_d \mid \forall c \in C_A : e_d = rd(c), e_d \cup E_d.$$

ідентифікований унікальним URI. ed визначається функцією [2,3]

Для верифікації в [7] було виділено поняття «функціональної одиниці системи» f - елементарної структурної складової інформаційної системи, що реалізує закінчений функціональний блок, для перевірки якого може бути розроблений один або більше автоматизованих або автоматичних перевірочних тестів:

$$F = (E_d, P, r) \mid \forall e_d, e_d \in E_d, \exists p, p \in P.$$

Функціональна одиниця розробки вибирається залежно від виду інформаційно-вимірювальної системи. Наприклад, якщо розробляється інформаційно-вимірювальна система, яка вимагає зв'язку з базою даних або звернення до віддаленого додатку, що має інший формат даних, Функціональна одиниця є функцією обміну інформацією з базою даних (віддаленим додатком). Ґрунтуючись на моделі верифікації інформаційних систем і онтологічної моделі

для адаптивного web середовища, які надають основні поняття і дозволяють їх пов'язати з вимогами до якості і засобами тестування, необхідно врахувати особливості інформаційно-вимірювальних систем з обмеженими ресурсами для створення відповідної моделі верифікації. Також необхідно розширити поняття функціональна одиниця з метою автоматизації процесу тестування не тільки програмного, а й апаратного забезпечення і засобів передачі даних між елементами інформаційно-вимірювальної системи з обмеженими ресурсами.

Особливість сучасних вбудованих систем полягає в тому, що архітектура системи теж стає адаптивною, для чого визначимо, що сукупність явних $UP(u)$ і неявних $UM(u)$ вимог користувача включає не тільки безліч елементів прототипу інтерфейсу, але і безліч вимог до елементів апаратного забезпечення. Безліч ресурсів приймемо за E , безліч адаптивних елементів апаратного забезпечення позначимо E_h , Тоді для адаптивної вбудованої системи будемо вважати

$$E = E_d \cup E_h, e \in E.$$

Для верифікації систем з веб-інтерфейсом візьмемо модель верифікації web-орієнтованих систем [9], яка побудована на підставі поняття концепції:

$$MV = (interprototype, E_d, DB, rmv),$$

$$rmv: interprototype \rightarrow E_d \times DB | \forall e_a, e_a \in \{rp(a),$$

$$rm(a)\}, e_a \rightarrow e, e = \{e_d\}.$$

Однак дана модель (1) описує лише структуру веб-інтерфейсу, не враховуючи взаємозв'язку системи з апаратним забезпеченням(яке має у складі RASPBERRY PI MODEL 3, що є елементом з обмеженим ресурсом), а також процес передачі даних між датчиками, лічильниками, логерами-реєстраторами та сервером. В існуючій моделі відсутній опис структури системи з обмеженими ресурсами для процесу верифікації та взаємозв'язок між елементами системи і засобами тестування.

Для розширення поняття функціональна одиниця з метою автоматизації процесу тестування апаратного забезпечення і засобів передачі даних між елементами інформаційно-вимірювальної системи з обмеженими ресурсами пропонується інтегрувати в (3) комплекс тестів по верифікації апаратного забезпечення $Ph = I, O, Os$.. Мета даного комплексу полягає в тестуванні методом «чорного ящика» коректності отриманих даних (O) від апаратного забезпечення у відповідь на відправлений запит, який містить якийсь множину вхідних даних (I). Отримана множина відповідей O порівнюється з еталонним значенням Os і з комунікаційним протоколом, на підставі їх відповідності приймається рішення щодо коректності функціонування апаратного забезпечення.

Відповідно до вищесказаного, множина функціональних одиниць інформаційно-вимірювальної системи з обмеженими ресурсами набуде вигляду:

$$F = (E_d, E_h, P, P_h, P_c, r) \mid \forall e_d, e_h \in E_d, \exists p, p \in P, \exists p_h, p_h \in P_h.$$

Для обчислення періоду передачі даних від i -го реєстратора інформаційно-вимірювальної системи скористаємося [8]:

$$\eta_i' = (1 + \omega_i)(\tau V \log_i + \sigma V err_i) \left(\frac{\tau}{V arch_i} - \frac{1}{v_{net}} \right). \quad (5)$$

Результати верифікації представимо у вигляді множини $\langle Y_{Log}, Y_{Res}, Y_{resp}, Y_{err} \rangle$. З огляду на зазначені переваги, модель для верифікації систем з обмеженими ресурсами MV набуде вигляду:

$$MV = (interprototype, E_d, E_h, DB, rmv, Y, \eta),$$

$$rmv : interprototype \rightarrow E_d \times E_h \times DB \mid \forall e_a, e_a \in \{rp(a),$$

$$rm(a)\}, e_a \rightarrow e, e = \{e_d\}.$$

ВИСНОВКИ

У цьому розділі наведено розробку моделі верифікації інформаційно-вимірювальної системи з обмеженими ресурсами. За результатами аналізу потрібно створити програмне забезпечення що відповідає наступним вимогам

- верифікація апаратного забезпечення;
- верифікація програмного забезпечення
- верифікуються дані, що надходять від апаратного забезпечення;
- верифікуються що відповідь від логера не перевищує заданий діапазон;
- надсилаються повідомлення про невдалу верифікацію користувачу контролеру.

Більш докладно вимоги до програмного забезпечення розкрито в додатку А.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДЕЛІ ВЕРИФІКАЦІЇ ІНФОРМАЦІЙНО ВИМІРЮВАЛЬНОЇ СИСТЕМИ

3.1 Ескізний проект

Актором системи є користувач контролер, він контролює стан логерів та лічильників, які є в будівлях його організації.

Для моделювання варіантів використання побудовано діаграму прецедентів, яку представлено на рисунку 3.1.

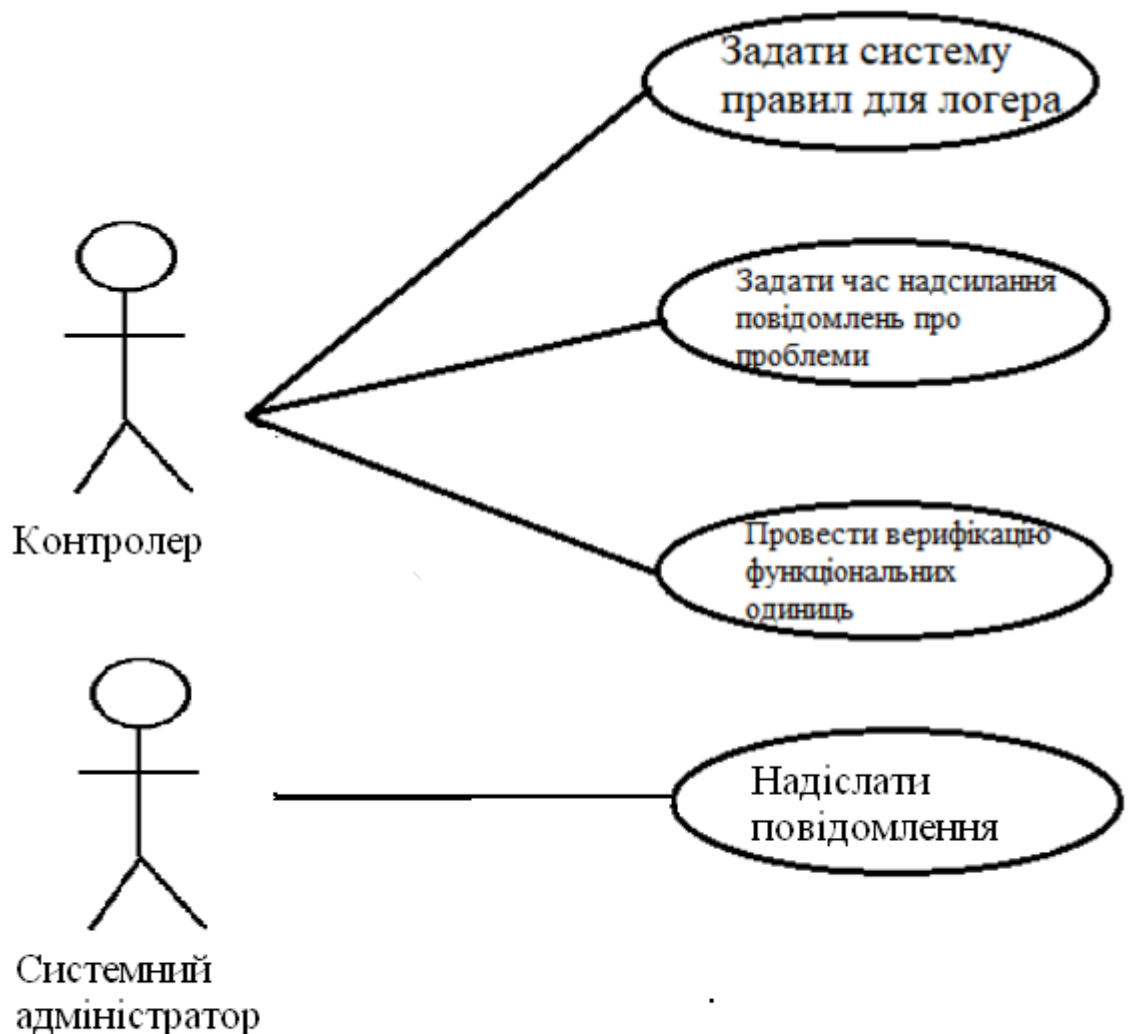


Рисунок 3.1 – Діаграма використання програмного забезпечення верифікації інформаційно-вимірювальної системи з обмеженими ресурсами

У таблиці 3.1 наведено виявлені варіанти використання інформаційно-вимірювальної системи з формулюванням основного змісту.

Таблиця 3.1 – Виявлені варіанти використання

Основні актори	Найменування	Формулювання
Контролер	Задати систему правил для логера	Дозволяє визначити діапазон даних, що є коректним для конкретної моделі і типу логгера, а також галузі де він використовується, а також задати коректний для цього типу логгера час відповіді на запит сервера. Після внесення даних користувач повинен мати можливість зберегти внесені зміни чи повернутися на попередню форму.
Контролер	Задати час надсилання повідомлень про проблеми	Дозволяє визначити критичний час для кожної ланки, що підключена до логера та встановити час за який контролера повинно бути повідомлено про проблеми, щоб це не призвело до великих проблем. Потрібно враховувати дні/робочі/вихідні та робочий/неробочий час
Контролер	Провести верифікацію функціональних одиниць	Дозволяє перевірити чи є відповіді від усіх апаратних функціональних одиниць, чи дані які вони надсилають є валідними, та сформулювати звіт з валідаційної сесії за проблемними ланками
Системний адміністратор	Надіслати повідомлення про помилки	Дозволяє наіслати як вручну так і перевірити автоматичне надсилання повідомлень про знайдені помилки контролерам інформаційно-вимірювальної системи

Конкретизуємо кожний варіант використання за допомогою діаграм діяльності.

Найменування “Задати систему правил для логера”

Основна дійова особа: Контролер

Зв'язок з іншими варіантами використання - відсутній

Короткий опис: Варіант використання “Задати систему правил для логера” дозволяє визначити діапазон даних, що є коректним для конкретної моделі і типу логера, а також галузі де він використовується, а також задати коректний для цього типу логера час відповіді на запит сервера. Після внесення даних користувач повинен мати можливість зберегти внесені зміни чи повернутися на попередню форму.

Початковий стан: прецедент починається коли вибраний логер та натиснуто кнопку редагувати його властивості.

Діаграму діяльності для цього прецеденту наведено на рисунку 3.2.

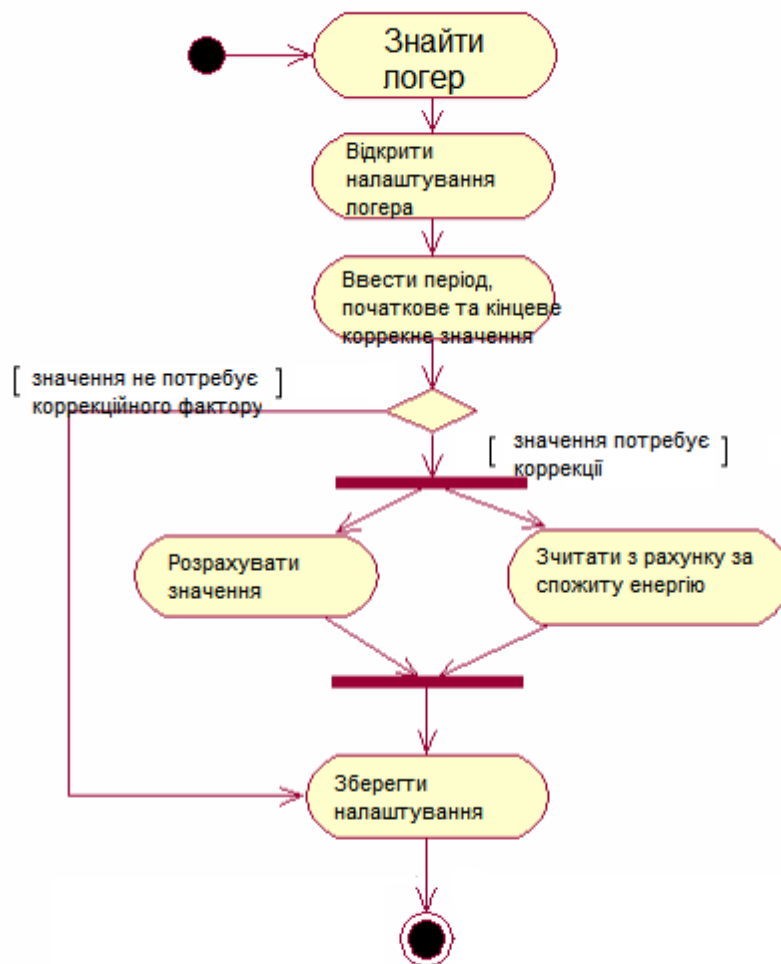


Рисунок 3.2 – Діаграма діяльності для варіанту використання “Задати систему правил для логера”

Найменування “Задати час надсилення повідомлень про проблеми”

Основна дійова особа: Контролер

Зв'язок з іншими варіантами використання - відсутній

Короткий опис: Варіант використання “Задати час надсилення повідомлень про проблеми” дозволяє визначити критичний час для кожної ланки, що підключена до логера та встановити час за який контролера повинно бути повідомлено про проблеми, щоб це не призвело до великих проблем. Потрібно враховувати дні/робочі/вихідні та робочий/неробочий час

Початковий стан: прецедент починається коли вибраний профіль користувача та натиснуто кнопку редагувати його налаштування.

Діаграму діяльності для цього прецеденту наведено на рисунку 3.3.

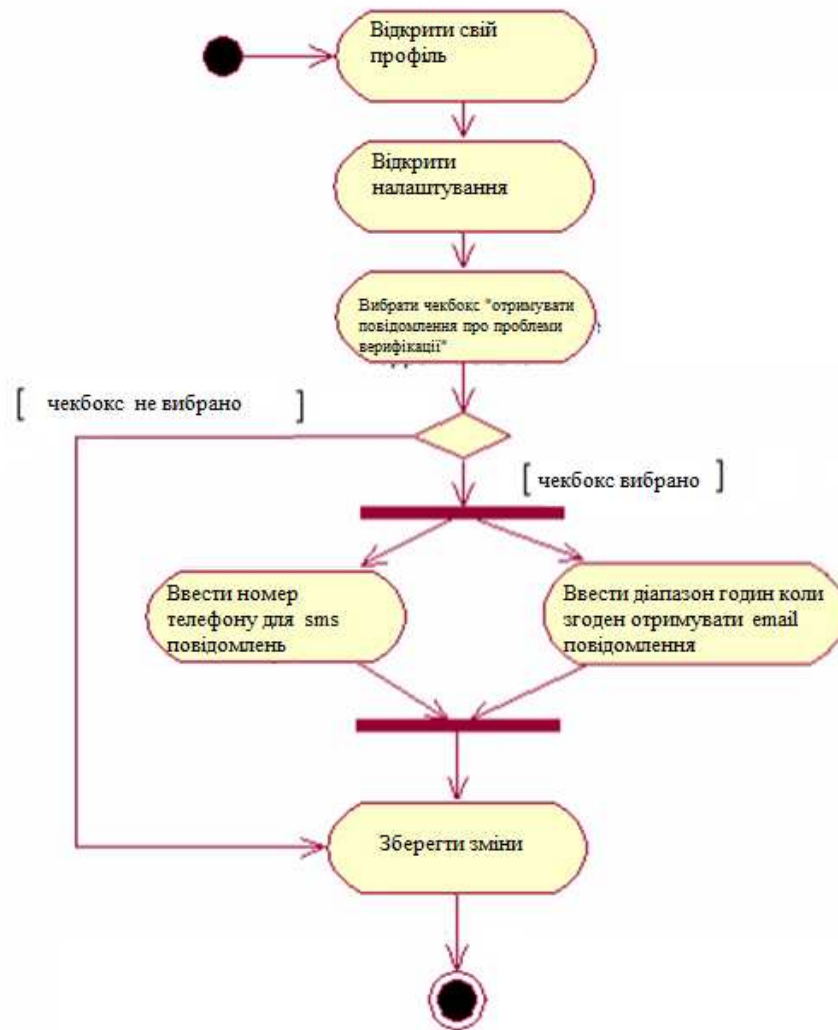


Рисунок 3.3 – Діаграма діяльності варіанта використання «Задати час надсилання повідомлень про проблеми»

Найменування “Провести верифікацію функціональних одиниць”

Основна дійова особа: Контролер

Зв'язок з іншими варіантами використання - відсутній

Короткий опис: Варіант використання “Провести верифікацію функціональних одиниць” дозволяє перевірити чи є відповіді від усіх апаратних функціональних одиниць, чи дані які вони надсилають є валідними, та сформуванати звіт з валідаційної сесії за проблемними ланками

Діаграму діяльності для цього прецеденту наведено на рисунку 3.4.

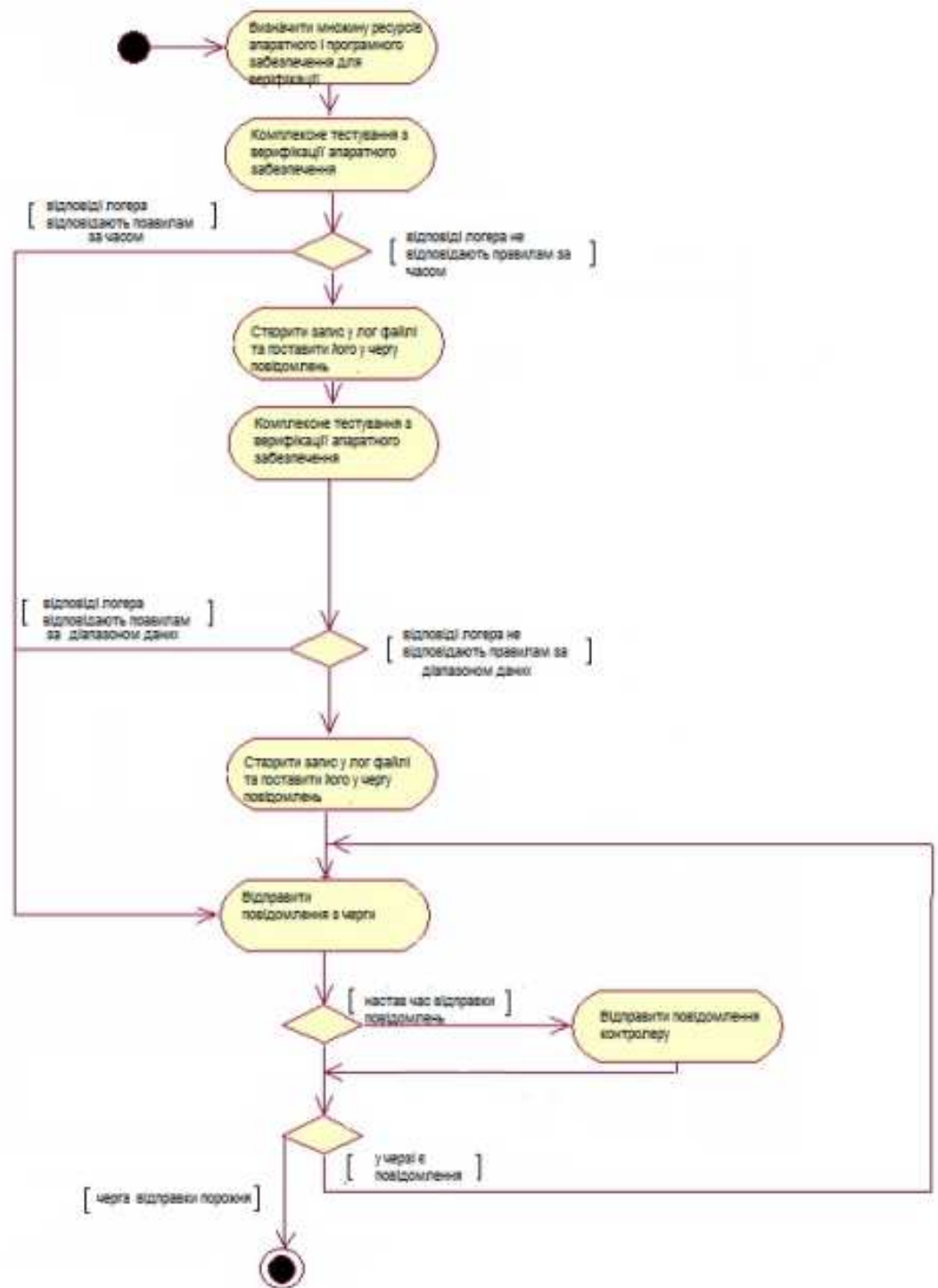


Рисунок 3.4 – Діаграма діяльності для варіанта використання «Провести верифікацію функціональних одиниць»

Найменування “Надіслати повідомлення контролерам”

Основна дійова особа: Системний адміністратор

Зв'язок з іншими варіантами використання - відсутній

Короткий опис: Варіант використання “ Надіслати повідомлення контролерам” дозволяє визначити критичний час для кожної ланки,що підключена до логера та встановити час за який контролера повинно бути повідомлено про проблеми, щоб це не призвело до великих проблем. Потрібно враховувати дні/робочі/вихідні та робочий/неробочий час

Початковий стан: прецедент починається коли вибраний профіль користувача та натиснуто кнопку редагувати його налаштування.

Діаграму діяльності для цього прецеденту наведено на рисунку 3.5.

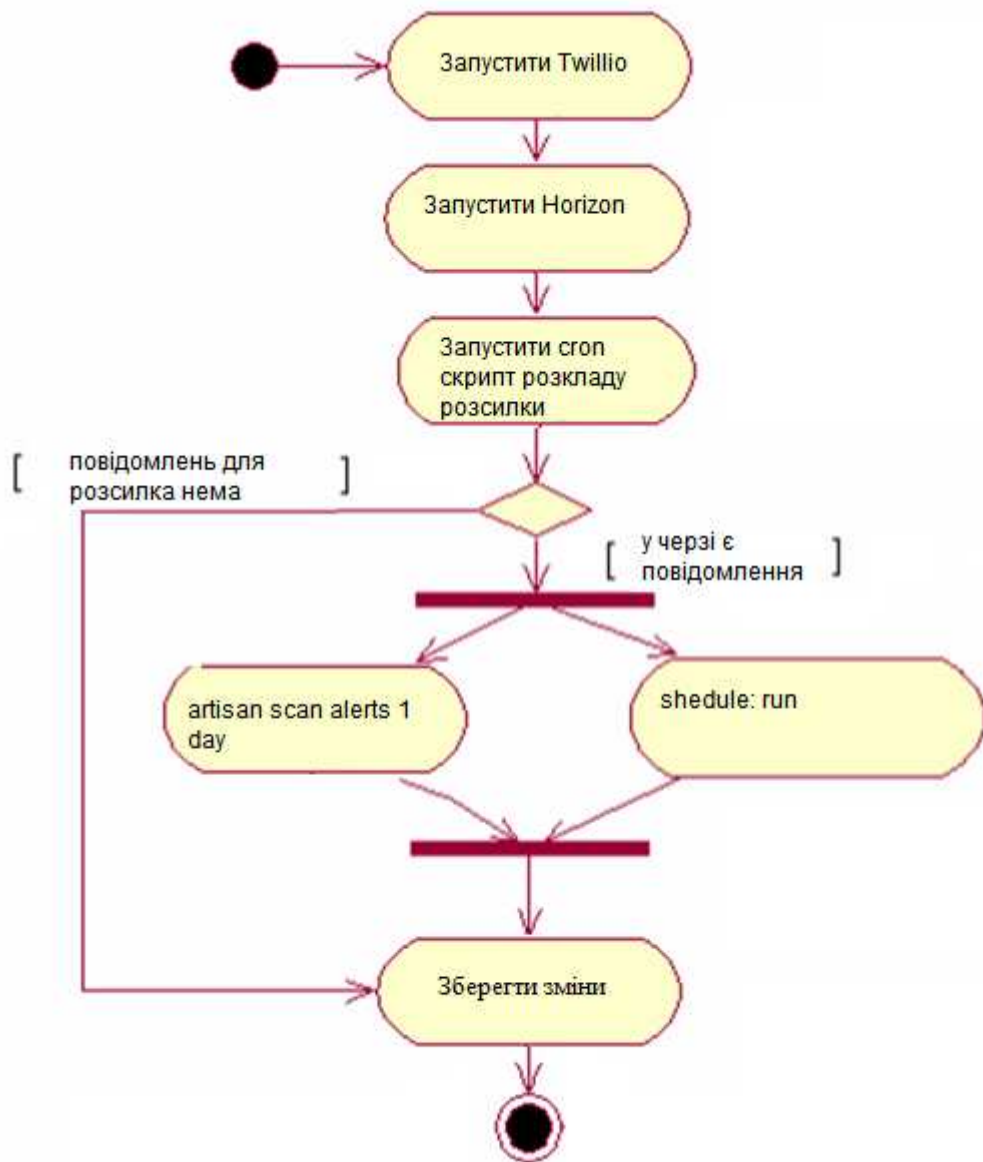


Рисунок 3.5 – Діаграма діяльності для прецеденту «Надіслати повідомлення про помилки»

3.2 Технічний проєкт

Вибір архітектурного стилю

Програмне забезпечення що розробляється найбільш відповідає архітектурному стилю MVVM, що полегшує відокремлення розробки графічного інтерфейсу від розробки бізнес логіки (бек-енд логіки), відомої як модель (можна також сказати, що це відокремлення представлення від моделі). Модель представлення є частиною, яка відповідає за перетворення даних для їх подальшої підтримки і використання. З цієї точки зору, модель представлення більше схожа на модель, ніж на представлення і обробляє більшість, якщо не всю, логіку відображення даних. Модель представлення може також реалізовувати патерн медіатор, організовуючи доступ до бекенд логіки навколо множини правил використання, які підтримуються представленням.

Архітектура програмного забезпечення у вигляді діаграми UML представлена на рисунку 3.6, а діаграма компонентів інформаційно-вимірювальної системи представлено на рисунку 3.7.

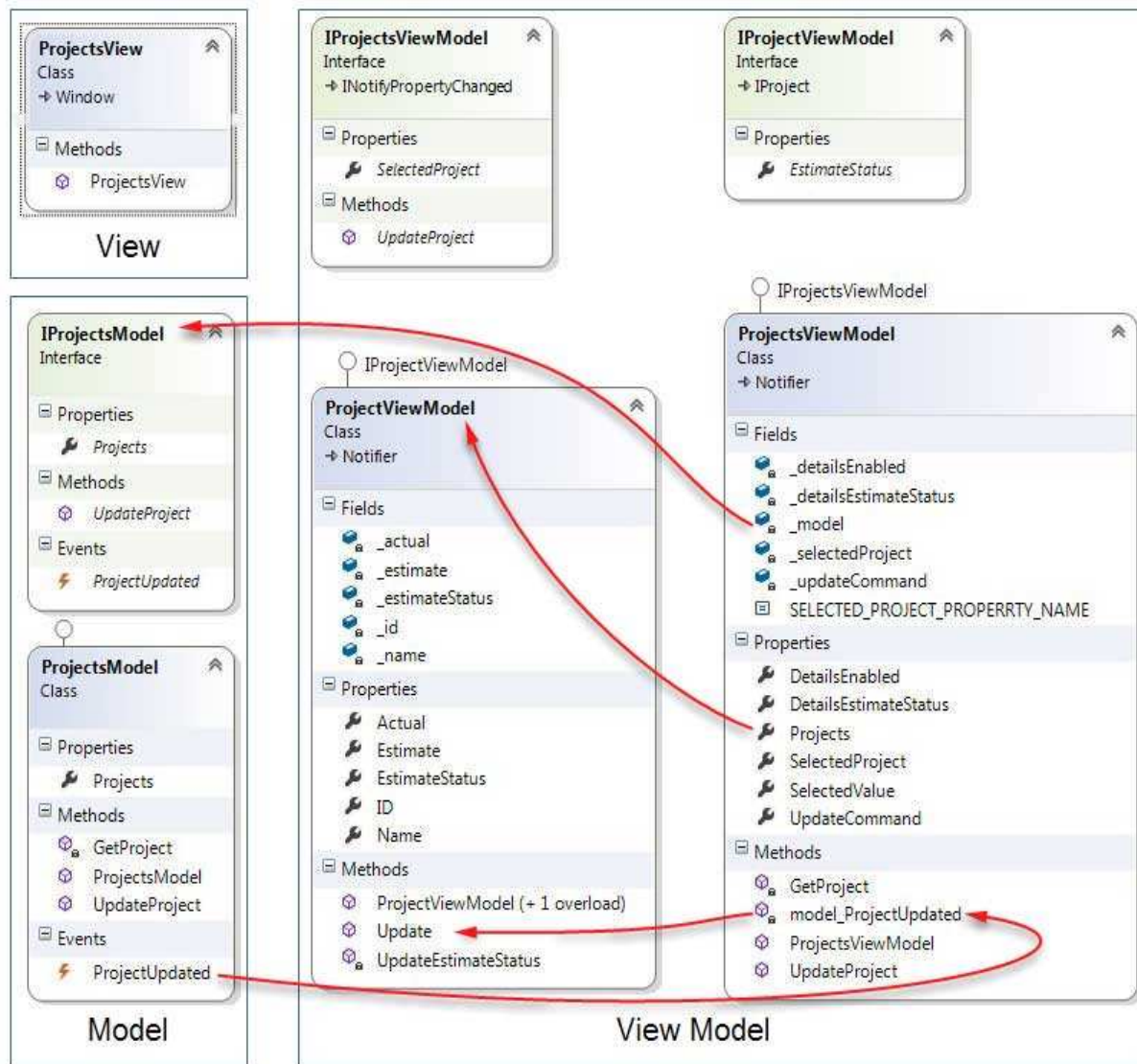


Рисунок 3.6 — Діаграма компонентів інформаційно-вимірювальної системи
(з указанням механізму роботи за шаблоном MVVC)

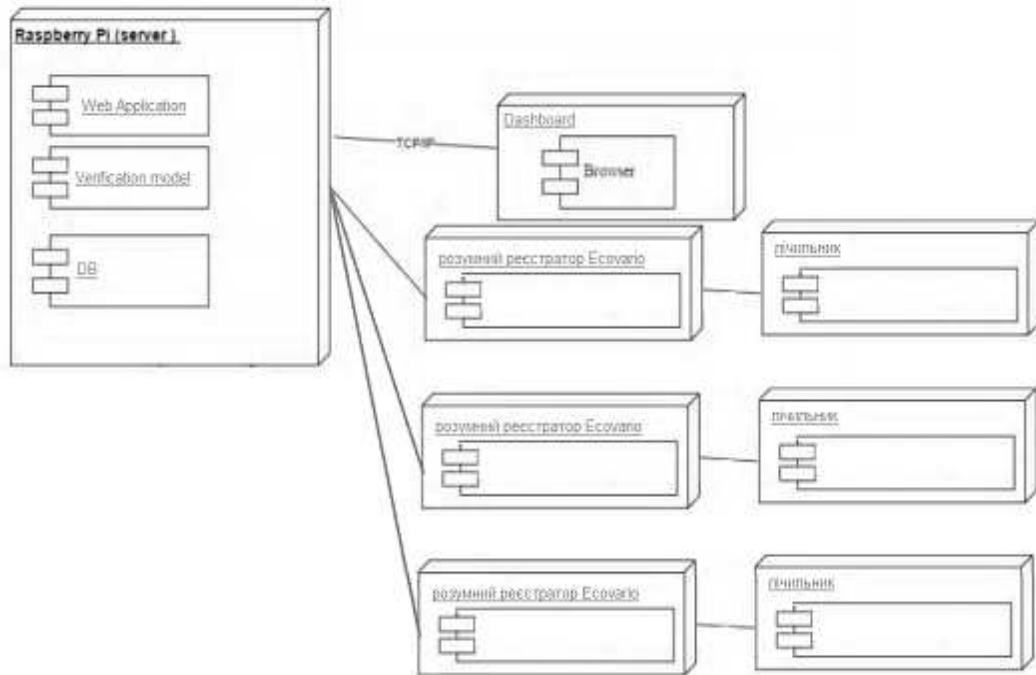


Рисунок 3.7 — Діаграма компонентів інформаційно-вимірювальної системи

Діаграму розгортання програмного забезпечення для верифікації інформаційно-вимірювальної системи представлено на рисунку 3.8.

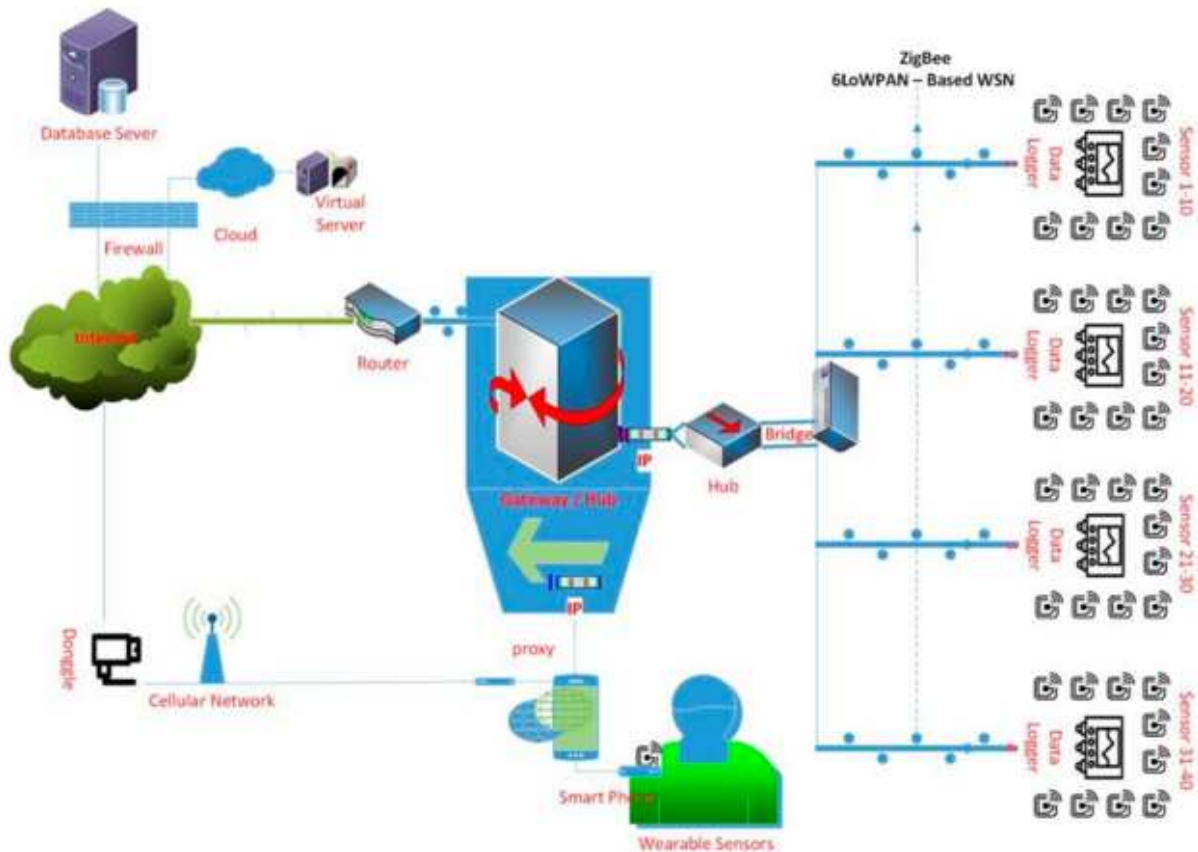


Рисунок 3.8 - Діаграма розгортання програмного забезпечення для верифікації інформаційно-вимірювальної системи

3.3. Робочий проект

Кодування програми проводиться на основі специфікації програмних модулів, представлених у підрозділі технічного проекту. Текст програми наведено у Додатку В.

ВИСНОВКИ

У третьому розділі проведено розробку проекту програмного забезпечення верифікації інформаційно-вимірювальної системи з обмеженими ресурсами.

4 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Результати застосування запропонованої моделі верифікації представлені у вигляді розробленого програмного забезпечення системи для верифікації вбудованих систем з обмеженими ресурсами (тестування проводилось на логгерах-реєстраторах до складу яких входить Raspberry Pi 3 Model B+), яка має можливість працювати автономно в різних середовищах, зберігати інформацію в базах даних, а також враховувати ієрархічну структуру підключених один до одного пристроїв, відображати результати верифікації, зберігати лог файл з результатами та надсилати push повідомлення користувачам, що відповідають за нагляд за цією ланкою інформаційно-вимірювальної системи.

Вирішення цього завдання досягається тим, що електронна інформаційна система для гнучкої верифікації вбудованих систем, має центральний системний блок управління базами даних, з'єднаний в єдину мережу з інформаційним сенсорним терміналом (1 he), аудіосистемою (2 he), мікрофоном (3 he), відеокамерою (4 he), пристроєм, реєструє зміну світлового потоку (5 he), модулем бездротового зв'язку, а також web-камерою (6 he), причому в якості центрального системного блоку управління базами даних використовується автономний одноплатний комп'ютер (7 he), який додатково містить блок синтезу даних для багатошарових вбудованих систем і модуль верифікації інформаційних систем з веб-інтерфейсом, а в як модуль бездротового зв'язку використовується WiFi модуль (8 he) і додатково містить GSM модуль (9 he). Для кожного із зазначених апаратних пристроїв існує програмне забезпечення, яке відповідає за прийом і обробку даних з фізичних пристроїв.

Для верифікації даної системи розроблено множину варіантів тестування P , де $rf: e \rightarrow p, f = \square \{e, p\}, e \square E, f \square F$.

Результати верифікації представлені в таблиці 4.1.

Як видно з результатів верифікації, пристрій 6 (Web-камера 6 he) не пройшов тестування програмного забезпечення, що пояснюється конфліктом

апаратно-програмного забезпечення 6 he і 4 he. Пристрій 9 he (GSM модуль) не пройшов тестування, через відмову програмного забезпечення.

Таблиця 4.1 - Результати верифікації інформаційно-вимірювальної системи

Номер ресурсу	Результати тестування		
	Апаратного забезпечення	Програмного забезпечення	Втрата даних при передачі
1	Тест пройдено	Тест пройдено	0 %
2	Тест пройдено	Тест пройдено	0 %
3	Тест пройдено	Тест пройдено	0 %
4	Тест пройдено	Тест пройдено	0 %
5	Тест пройдено	Тест пройдено	0 %
6	Тест не пройдено	Тест пройдено	0 %
7	Тест пройдено	Тест не пройдено	0 %
8	Тест не пройдено	Тест не пройдено	100%

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВІД РОЗРОБКИ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Дана робота присвячена розробці програмного забезпечення верифікації інформаційно-вимірjuвальних систем. З ростом ринку програмного забезпечення для розумних будинків цей напрямok актуальний як ніколи.

Створення програмного забезпечення вимагає одноразових витрат на розробку, придбання необхідних технічних засобів і поточних витрат на функціонування продукту. Економія від функціонування ПЗ визначається з урахуванням витрат на його експлуатацію. Відношення цієї економії до витрат на створення програмного забезпечення характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за діючими на момент розрахунку оптовими цінами, тарифами і ставками заробітної плати.

5.1 Розрахунок витрат на створення й експлуатацію програмного забезпечення

Витрати на розробку системи складаються з витрат:

- на заробітну платню розробника;
- на амортизацію ЕОМ, на якій виконується розробка;
- на експлуатацію цієї ЕОМ;
- на засоби розробки;
- на матеріали і комплектуючі.

Розробка програмного забезпечення виконується програмістом, місячна заробітна плата якого складає 8000 грн.

Вартість сучасного ПК становить 15000 грн.

При вартості кіловат-години електроенергії 1.68 грн, розраховується вартість розробки програми.

Витрати на допоміжні матеріали наведені в табл. 5.1.

Таблиця 5.1 – Витрати на допоміжні матеріали

№	Пункти витрат	Сума, грн
1	Флеш-накопичувач даних	250
2	Папір для принтера	80
3	Картридж та тонер для принтера	400
	Разом	730

Вартість ПЗ знаходиться за формулою:

$$C_{\text{пр}} = (З_{\text{зп}} + З_{\text{сз}} + З_{\text{зг}} + З_{\text{е}}) * T + З_{\text{м}} \quad (4.1)$$

де T – тривалість розробки, міс.

$З_{\text{зп}}$ – основна і додаткова заробітна плата обслуговуючого персоналу, грн.

$З_{\text{сз}}$ – відрахування на соціальні заходи (15% від основної і додаткової заробітної плати), грн.

$З_{\text{зг}}$ – загальногосподарські витрати (10% від основної заробітної плати), грн.

$З_{\text{е}}$ – витрати на електроенергію, грн.

$З_{\text{м}}$ – витрати на основні і допоміжні матеріали, грн.

$З_{\text{е}}$ при споживанні потужності 0.5 кВт, тривалості роботи на місяць, рівної $22*8 = 176$ годин, вартості кіловат-години електроенергії 1.29 грн становить:

$$З_{\text{е}} = 176 * 1.68 * 0.5 = 147.84 \text{ грн}$$

Витрати на розробку ПЗ наведені в табл. 5.2.

Таблиця 5.2 – Витрати на розробку програмного забезпечення

№	Найменування витрат	Одиниця	Кількість
1	Тривалість розробки (Т)	Міс.	3
2	Основна і додаткова заробітна плата (З _{зп})	Грн.	8000.00
3	Відрахування на соціальні заходи (З _{сз})	Грн.	1200.00
4	Загальногосподарські витрати (З _{зг})	Грн.	800.00
5	Витрати на допоміжні матеріали (З _м)	Грн.	730.00
6	Витрати на електроенергію (З _е)	Грн.	147.84

За формулою 1 розрахуємо вартість програми:

$$C_{\text{пр}} = (8000 + 1200 + 800 + 147.84) * 3 + 730 = 31173.52 \text{ грн}$$

Амортизаційні відрахування складають 25% балансової вартості на рік:

$$A_{\text{об}} = 15000 * 0.25 = 3750 \text{ грн}$$

В масштабах підприємства річні витрати на основні і допоміжні матеріали визначаються в розмірі 5% вартості основного устаткування:

$$B_{\text{м}} = 15000 * 0.05 = 750 \text{ грн}$$

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$\Phi_{\text{м}} = 264.5 * T_3 \text{ (5.2)}$$

T_3 – це середнє місячне навантаження устаткування (близько 6 годин), 264.5 – середня кількість робочих днів на рік.

Таким чином, річний обсяг роботи ПК складає:

$$\Phi_m = 264.5 * 6 = 1587 \text{ годин}$$

Витрати на електроенергію $З_e$ при 1587 годинах роботи устаткування на рік становлять:

$$З_e = 1587 * 1.68 * 0.5 = 1333.08 \text{ грн}$$

Експлуатаційні витрати на ПК на рік складуть:

$$З_{зр} = 3750 + 750 + 1333.08 = 5833.08 \text{ грн}$$

Отже, у перший рік витрати на створення й експлуатацію ПЗ складуть:

$$З_p = 31173.52 + 5883.08 = 37006.6 \text{ грн}$$

5.2 Економічна ефективність розробки і впровадження програмного забезпечення для балансування обчислювального навантаження

Основним показником економічної ефективності програмного забезпечення є зменшення трудових витрат та часу на обробку завдання, а також ефективне розподілення трудових ресурсів.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження програмного продукту вивільняє 1.5 працівника (за експертною оцінкою фахівців підприємства).

Заробітна плата 1.5 працівника в рік складає:

$$З = 8000 * 12 * 1.5 = 144000 \text{ грн}$$

Річний економічний ефект розраховується за формулою:

$$E_{\text{рік}} = \Delta C_n - E_n * k \quad (5.3)$$

де ΔC_n – вивільнені кошти після впровадження продукту (144000 грн) мінус експлуатаційні витрати (37006.6 грн) = 106993.4 грн;

E_n – коефіцієнт ефективності (дорівнює коефіцієнту амортизації 0.25);

k – одноразові витрати на впровадження продукту (30870.56 грн).

$$E_{\text{рік}} = 144000 - 0.25 * 31173.52 = 136206.62 \text{ грн}$$

Термін окупності ПЗ розраховується за формулою:

$$T = k / \Delta C_n = 31173.52 / 106993.4 = 0.29 \text{ року}$$

Отже, термін окупності ПЗ становить приблизно 4 місяці.

Висновки

У цьому розділі були здійснені розрахунки витрат, економічної ефективності та терміну окупності ПЗ для верифікації даних енергетичної інформаційно-вимірювальної системи. Виходячи з розрахунків, впровадження даного програмного забезпечення окупить себе протягом 4 місяців. Таким чином, його розробка є економічно обґрунтованою.

6 ОХОРОНА ПРАЦІ

Вступ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних мір і засобів, спрямованих на збереження здоров'я й працездатності людини в процесі праці (ст. 1 Закону „Про охорону праці”).

Її завданням є зведення до мінімуму ймовірності травмування або захворювань працюючих з одночасним забезпеченням нормальних умов праці при його максимальній продуктивності, науковий аналіз умов праці, виробничих процесів та встаткування з погляду можливих аварійних ситуацій, поява небезпечних факторів, виділення шкідливих речовин. На основі такого аналізу визначаються небезпечні ділянки виробництва, можливі аварійні ситуації, розробляються заходи щодо їхнього запобігання або обмеження наслідків.

Сьогодні комп'ютерна техніка широко застосовується у всіх сферах людської діяльності. Усе більше людей різних професій не можуть обійтися без допомоги комп'ютера. Це не дивно, оскільки розрахунки за допомогою ЕОМ значно допомагають заощадити час і кошти. Тому, необхідно приділяти більше уваги забезпеченню безпечних і нешкідливих умов праці користувачів ЕОМ, підвищити контроль над підтримкою діючих норм, стандартів, правил, інструкцій та інших офіційних документів по техніці безпеки споруджень, обладнання та машин.

Під час роботи на комп'ютері людина піддається впливові ряду шкідливих і небезпечних факторів, що пов'язано з небезпекою одержання травм і професійних захворювань.

Максимально зменшити кількість шкідливих впливів на людину при високій продуктивності праці, створити комфортні умови для роботи людей – ось

одна з головних задач охорони праці.

Умови праці користувачів ЕОМ повинні відповідати закону «Про охорону праці» , який визначає основні положення, спрямовані на реалізацію прав громадян, охорону їхнього життя і здоров'я в процесі їхньої трудової діяльності, регулює відносини між власником підприємства і робітником з питань безпеки гігієни праці, установлює єдиний порядок організації праці на Україні.

6.2 Аналіз небезпечних і шкідливих факторів

При роботі з ПК людина може піддатися впливу шкідливих та небезпечних факторів. Під шкідливими виробничими факторами розуміють фактори, тривалий вплив яких викликає розвиток професійних захворювань. Небезпечні виробничі фактори – вплив яких на працюючого викликає травму, тобто пошкодження організму. З точки зору впливу небезпечних і шкідливих факторів при роботі з ПК можна виділити наступні:

- недостатня освітленість робочого місця;
- вплив електромагнітного й електростатичного випромінювання;
- вплив електричного струму;
- вплив шуму;
- несприятливі умови мікроклімату;
- нераціональне розташування устаткування і неправильна організація робочого місця.

Розглянемо більш детально вплив цих факторів на користувача ПК.

6.2.1 Недостатня освітленість робочого місця

Напружена зорова робота внаслідок нераціонального освітлення може бути причиною функціональних порушень у зоровому аналізаторі і привести до розладу зору, а у важких випадках – і до повної втрати.

Втома органів зору залежить від ступеня напруженості процесів, що супроводжують зорове сприйняття. При роботі користувача ПК одним з основних навантажень на організм є навантаження на зір, тому дуже важливо приділити особливу увагу освітленості робочого місця.

У залежності від джерела світла виробниче освітлення може бути трьох видів: природним, штучним, сполученим.

При роботі з ПК вдень у ясну погоду використовується природне освітлення, яке здійснюється через світлові віконні прорізи. При нестачі природного світла, а також при роботі у вечірній час використовується штучне освітлення, джерелами якого є газорозрядні лампи. При такому освітленні величезне значення має правильне розташування джерел світла, їхній тип і кількість.

Правильна освітленість робочого місця сприяє високій працездатності користувача ПК і зводить до мінімуму шкідливий вплив на органи зору. Освітленість робочого місця, що рекомендується для користувача ПК, складає 400 - 750 лк.

6.2.2 Вплив електромагнітного й електростатичного випромінювань

При роботі ПК виникають два типи випромінювання: електростатичне й електромагнітне. Навколо електростатичного зарядженого монітора підвищується концентрація пилу. Електромагнітне випромінювання створюється магнітними котушками, що знаходяться біля цокольної частини ЕПТ. Результати медичних

досліджень показують, що електризований пил викликає запалення шкіри, а електромагнітне випромінювання приводить до порушення біологічних процесів (порушення сил, зниження кров'яного тиску, уповільнення серцевої діяльності, зміни складу крові).

6.2.3 Вплив електричного струму

Робота комп'ютерів і периферійних пристроїв можлива завдяки електричному струмові, до джерел якого вони підключаються, тому існує небезпека ураження користувача струмом. Щоб уникнути нещасних випадків варто строго дотримуватись правил електробезпеки.

Небезпека електричного струму на відміну від інших небезпек збільшується тим, що людина не в змозі без спеціальних приладів виявити напругу дистанційно, а також швидкоплинністю поразки – небезпека виявляється, коли людина вже уражена. Ураження струмом приводить до різних порушень в організмі, викликаючи як місцеву поразку тканин і органів, так і загальне ураження організму.

Людина відчуває дратівну дію перемінного струму промислової частоти силою 0,6 – 1,5 мА і постійного струму 5 -7 мА. Ці струми не представляють серйозної небезпеки для організму людини, тому що при такій силі струму можливе самостійне звільнення людини від контакту зі струмоведучими частинами. Для перемінного струму промислової частоти сила невідпускаючого ефекту, але приводить до сильних болючих відчуттів, сила такого струму 15 – 80 мА і більше.

6.2.4 Вплив шуму

Шумом прийнято називати всякий небажаний для людини звук, що заважає сприйняттю корисних сигналів. Шум являє собою безладне сполучення звуків різної інтенсивності і частоти.

Вплив шуму на організм людини залежить від рівня звукового тиску, частотних характеристик, тривалості дії, а також індивідуальних особливостей людини.

Робота ПК може супроводжуватися шумами, викликаними деякими рухливими пристроями усередині комп'ютера, а також підключеними до нього. До таких пристроїв можна віднести вентилятори, пристрої читання дисків, принтери. При тривалій дії шуму у оператора ЕОМ виявляються симптоми утомленості, нервового збудження, що сприяють погіршенню працездатності і допущенні помилок при роботі.

6.2.5 Несприятливі умови мікроклімату

Мікроклімат – метеорологічні умови внутрішнього середовища приміщень, які визначаються діючими на людину сполученнями температури, відносної вологості, швидкості руху повітря й теплового випромінювання. Мікрокліматичні параметри впливають на самопочуття і здоров'я людини, а також на надійність роботи засобів обчислювальної техніки.

При підвищеній температурі і вологості в приміщенні відбувається перегрівання тіла, що погіршує самопочуття і працездатність. Варто враховувати, що ПК, які знаходяться в приміщенні, сприяють збільшенню температури.

Для комфортної роботи приймається температура $23 (\pm 2)^{\circ}\text{C}$ і $18 (\pm 2)^{\circ}\text{C}$ в теплий і холодний час року відповідно, при відносній вологості $55 (\pm 5)$.

6.2.6 Нераціональне розташування устаткування і неправильна організація робочого місця

Робоче місце – це частина приросту, у якій користувач ПК здійснює трудову діяльність і проводить велику частину робочого часу. Важливу роль грає планування робочого місця, яке повинне задовольняти вимогам зручності виконання робіт і економії енергії і часу користувача, зручності обслуговування ЕОМ, дотримання правил охорони праці. При плануванні робочого місця необхідно врахувати зони досяжності при розташуванні моніторів, клавіатури, маніпулятора типу «миша». Ці зони встановлюють на підставі антропологічних даних людського тіла.

Під час роботи користувач ПК не повинен піддаватися впливові сторонніх подразників, якими можуть бути похмуре фарбування пристроїв ЕОМ і приміщення.

6.3 Розрахунок системи вентиляції приміщень на підприємстві

6.3.1 Проведення розрахунків загальнообмінної вентиляції виробничих приміщень

Повітрообмін - процес заміни відпрацьованого і забрудненого повітря у виробничому приміщенні свіжим за допомогою природної і (або) механічної вентиляції. Величина обміну повітря, м/год, - один з головних показників, необхідних для проектування будь-якої з систем вентиляції виробничого приміщення.

Вентиляційні системи загальнообмінної механічної вентиляції бувають витяжними, припливними і припливно-витяжними.

Обмін повітря повинен розраховуватись так, щоб концентрація шкідливостей (пари, газу, волога, пил або аерозолі та ін.) в приміщенні під час роботи вентиляції не перевищувала допустимих рівнів чи гранично допустимих концентрацій.

Кількість повітря, що подається у приміщення для забезпечення необхідних метеорологічних умов та якості повітря робочої зони, визначають відповідно до існуючих будівельних норм. Розрахунок здійснюють за такими чинниками: шкідливі речовини, надлишки теплоти, надлишки вологи, за кількістю працюючих. Якщо у приміщенні мають місце одночасно декілька шкідливих чинників, розрахунок ведуть за кожним із них і більше з отриманих значень приймають за розрахункове.

При одночасному надходженні у повітря приміщення кількох шкідливих речовин односпрямованої дії об'єми повітря, необхідні для розбавлення кожної речовини окремо, сумують, а при речовинах різноспрямованої дії - допускається приймати найбільший з розрахованих об'ємів.

Наприклад, при одночасному випаровуванні декількох розчинників (ацетону, спиртів, оцтової кислоти) чи подразнюючих газів (сірчаного чи сірчистого ангідриду, хлористого та фтористого водню та ін.) розрахунок кількості повітря для загальнообмінної вентиляції слід виконувати, сумуючи обсяги його, потрібні для розбавлення до норми окремо кожного з розчинників чи подразнюючих газів.

При одночасному виділенні кількох газів і парів, окрім розчинників і подразнюючих газів, кількість повітря при розрахунку вентиляції приймається за тією шкідливістю, яка потребує найбільшого розбавлення.

У виробничих приміщеннях, де виділяється значна кількість шкідливих газів, парів, пилу витяжка повинна бути на 10% більшою ніж приплив, щоб шкідливі речовини не витіснялись у суміжні приміщення з меншою шкідливістю.

В системі припливно-витяжної вентиляції можливе використання не лише

зовнішнього повітря, але й повітря самих приміщень після його очищення. Таке повторне використання повітря приміщень називається рециркуляцією і здійснюється в холодний період року для економії тепла, витраченого на підігрівання припливного повітря. Однак можливість рециркуляції обумовлюється цілою низкою санітарно-гігієнічних та протипожежних вимог.

Штучна (механічна) вентиляція, на відміну від природної, дає можливість очищувати повітря перед його викидом в атмосферу, вловлювати шкідливі речовини безпосередньо біля місць їх утворення, обробляти припливне повітря (очищувати, підігрівати, зволожувати тощо), більш цілеспрямовано подавати повітря в робочу зону. Окрім того, механічна вентиляція дає можливість організувати повітрозабір у найбільш чистій зоні території підприємства і навіть за її межами.

При штучній вентиляції повітрообмін здійснюється внаслідок різниці тисків, що створюється вентилятором. Вона застосовується в тих випадках, коли тепловиділення у виробничому приміщенні недостатні для постійного (протягом року) використання аерації, або коли кількість чи токсичність шкідливих речовин, які виділяються у повітря приміщення є такою, що виникає необхідність постійного повітрообміну незалежно від метеорологічних умов навколишнього середовища.

Механічна вентиляція може бути робочою або аварійною. Останню необхідно передбачати у виробничих приміщеннях, де можливе раптове надходження у повітря значної кількості шкідливих чи вибухонебезпечних речовин. Аварійна вентиляція повинна вмикатись автоматично у разі досягнення граничної концентрації небезпечних виділень і забезпечувати швидке їх вилучення із приміщення. Як правило, аварійна вентиляція повинна забезпечувати 8-12-разовий повітрообмін за годину в приміщенні.

6.3.2 Розрахунок вентиляції приміщень на підприємстві з виготовлення меблів

Вентиляція – це організований і регульований повітрообмін, що забезпечує видалення із приміщення забрудненого повітря і подачу на його місце свіжого.

Задачею вентиляції є забезпечення чистоти повітря і параметрів мікроклімату у відповідності з нормами.

Визначення необхідного повітрообміну у випадку загальнообмінної вентиляції в залежності від конкретних умов може бути визначений різними способами.

Підприємство має чотири приміщення – склад, два приміщення для обробки матеріалів, одна кімната для фарбування виробів.

Розрахунок необхідного повітрообміну за відсутності шкідливих речовин (шкідливі речовини, волога, надлишки тепла) проводиться за формулою: $L_s = n \cdot L$ де n – кількість працюючих; L – витрата повітря на одного працюючого.

При об'ємі приміщення на одного працюючого $V < 20 \text{ м}^3$ повітрообмін $L = 30 \text{ м}^3/\text{год}$ на одного працюючого; при $V > 20 \text{ м}^3$ $L = 20 \text{ м}^3/\text{год}$; при $V > 40 \text{ м}^3$ допускається природна вентиляція.

Таким чином, розрахуємо повітрообмін на складі та в приміщеннях з обробки сировини, де шкідливі речовини не використовуються.

Об'єм приміщення 1 $V_1 = 15 \cdot 10 \cdot 5 = 750, \text{ м}^3$, – допускається природна вентиляція.

Об'єм приміщення 2 $V_2 = 5 \cdot 7 \cdot 5 = 175, \text{ м}^3$, – допускається природна вентиляція.

Об'єм приміщення 3 $V_3 = 10 \cdot 6 \cdot 5 = 300, \text{ м}^3$, – допускається природна вентиляція.

ВИСНОВКИ

За результатами проведених розрахунків допускається природна вентиляція приміщень, наявних на підприємстві з виготовлення корпусних меблів. Це виправдано тим, що приміщення мають великі розміри та можуть провітрюватися без допоміжних штучних засобів.

7 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

7.1 Вступ

Навколишнє (природне) середовище – це сукупність об'єктів та умов природи, в яких проходить діяльність будь-якого суб'єкта, що відрізняються від інших складових частин навколишнього середовища властивостями самопідтримки та саморегуляції без впливу людини.

«Охорона навколишнього середовища» (ОНС) – це система законодавчих актів, нормативних документів, інструкцій, стандартів, ДСТ, що регламентують чи відображають вимоги до охорони природи та навколишнього середовища, при будівництві, реконструкції та експлуатації промислових об'єктів, будинків і споруджень, у тому числі житло-цивільного невиробничого призначення.

«Охорона навколишнього середовища» полягає в здійсненні комплексних технічних рішень по:

- охороні атмосферного повітря від забруднення;
- охороні поверхневих і підземних вод від забруднення, виснаження;
- відновленню (рекультивація) земельних ділянок;
- використанню родючого шару ґрунту, охороні надр і тваринного світу.

Наука і техніка початку третього тисячоліття розвивається в темпах геометричної прогресії, не є виключенням і промисловість як одна із самих (якщо не самої) масштабних сфер діяльності людини. Подібного роду тенденція поширилась по усьому світі і вже захопила, що розвиваються, у минулому слаборозвинені країни. У зв'язку з не бездоганністю технологічних процесів на даному етапі неминуче негативний вплив промисловості на навколишнє середовище, промислових відходів як компонента даного впливу. Щорічно в усьому світі й у нашій країні мільярди тонн твердих, рідких, газоподібних

відходів надходить і біосферу, завдаючи тим самим непоправної втрати як живій, так і неживій природі.

Зростаючі темпи зміни середовища проживання приводить до порушення взаємозв'язку між ними і людиною, зниженню адаптаційних можливостей організму. Середовище проживання може містити такі речовини, з якими організм в ході еволюції не зштовхувався і тому не має відповідних аналізаторних систем, що сигналізують про їхню залежність. У зв'язку з цим оцінити стан здоров'я людини, зрозуміти характер патології у відриві від аналізу змін, що відбуваються у навколишньому середовищі неможливо.

Велике значення тому має організація інформаційної системи «здоров'я населення – навколишнє середовище», дані для якої можуть збиратися через державну статистичну звітність. Задача державної інформаційної системи полягає у зборі даних про забруднення навколишнього середовища, стан здоров'я населення.

7.2 Забруднення навколишнього підприємством з виготовлення меблів

Забруднення – це наявність у навколишньому середовищі шкідливих речовин, що порушують функціонування екологічних систем або інших окремих елементів і знижують якості середовища з огляду проживання людини або ведення й господарської діяльності. Цим терміном характеризують всі тіла, речовини, явища, процеси, що з'являються в даному місці навколишнього середовища, але не в той час і не в тій кількості, які характерні для природи і можуть виводити її системи зі стану рівноваги.

На сучасному етапі розвитку суспільства поводження з відходами водночас з іншими екологічними проблемами посідає одне з чільних місць у системі екологічної безпеки і забезпечення сталого розвитку країни. Їхнє вирішення пов'язано з необхідністю узгодити комплекс екологічних, економічних і

соціальних завдань і вимагає постійних системних зусиль з боку органів управління, науковців та громадськості.

Підприємства деревообробної галузі, порівняно з підприємствами чорної металургії, хімічної, нафтодобувної, вугільної промисловості, справляють менший негативний вплив на довкілля, проте і надалі він досить відчутний.

На промислових підприємствах утворюється значна кількість відходів – як промислових, так і побутових, як твердих, так і рідких. Відходи, попри переважно несприятливий вплив на навколишнє природне середовище, створюють низку організаційно-технічних проблем характеру для підприємств. Відсутність на підприємстві спеціалістів з охорони навколишнього природного середовища унеможливорює належну організацію роботи щодо захисту довкілля загалом, і у сфері поводження зокрема. Як наслідок – низка негативних наслідків для самого підприємства: від стягнень, які накладаються зі сторони спеціально уповноважених органів державного нагляду за охороною навколишнього природного середовища як на посадових осіб підприємства, так і на саме підприємство, і аж до призупинення виробничої діяльності підприємства.

7.3 Заходи щодо зменшення забруднення навколишнього середовища підприємством

В роботі з оздоровлення навколишнього середовища, обмеження впливів шкідливих речовин на біосферу важливу роль грають служби контролю стану природи, середовища проживання людей, локального і регіонального моніторингу навколишнього середовища.

Ці служби, озброєні сучасною вимірювальною технікою і приладами контролю, повинні оперативно оповіщати населення про всі випадки наближення параметрів навколишнього середовища до небезпечного рівня.

При будівництві, введенні в експлуатацію, реконструкції і технічному

переозброєнні підприємств, а також при впровадженні нових технологічних процесів повинні передбачатися заходи для уловлення, знешкодження шкідливих речовин, зниження або повного припинення забруднюючих викидів в атмосферу. Необхідно дотримувати технічні нормативи і нормативи гранично допустимих норм викидів, виходячи з того, що сукупність викидів від спроектованих, діючих і запланованих до будівництва підприємств не повинна погіршувати якість атмосферного повітря. Здійснення заходів щодо охорони атмосферного повітря не повинно приводити до забруднення інших об'єктів природного середовища.

Щодо відходів треба вести планомірну повсякденну і всебічну політику як на рівні держави, так і на рівні кожного підприємства. Основним завданням є забезпечення мінімального утворення відходів, розширення їх використання у господарській діяльності, запобігання шкідливому впливу відходів на природу та здоров'я людини.

Згідно із Законодавством України про відходи – відходи є об'єктом права власності. Суб'єктами права власності на відходи є громадяни України, іноземці, особи без громадянства, підприємства, установи та організації усіх форм власності, територіальні громади, Автономна Республіка Крим і держава. Суб'єкти права власності володіють, користуються і розпоряджаються відходами в межах, визначених законом.

Відповідно до Закону України «Про відходи», поводження з відходами – дії, спрямовані на запобігання утворенню відходів, їх збирання, перевезення, зберігання, оброблення, утилізацію, видалення, знешкодження і захоронення, втому числі контроль за цими операціями та нагляд за місцями видалення.

Законом передбачено два альтернативних напрями поводження з відходами: утилізація (використання відходів як вторинних матеріальних чи енергетичних ресурсів) та видалення (в основному шляхом захоронення та знищення або знешкодження).

Підприємства лісопромислового комплексу зайняті заготівлею, механічною та хімічною переробкою деревини та інших лісових багатств. Основними його галузями є лісове господарство, власне лісова промисловість або лісозаготівельна, а також деревообробна (виробництво пиломатеріалів, фанери, сірників, столярних виробів, меблів), целюлозно-паперова (виробництво целюлози, паперу, картону) і лісохімічна (виробництво каніфолі, скипидару, спирту, смол тощо).

Підприємства деревообробної промисловості розміщені більш рівномірно на території країни, що спричинено впливом різних чинників на розміщення підприємств окремих підгалузей.

Основними джерелами забруднення атмосферного повітря на даному підприємстві з виготовлення меблів є: формальдегід, пара розчинників і розріджувачів, оксид азоту, анілін, азот, уайт-спірит, скипидар, аміак, деревний та лакофарбовий пил, Деревний пил, оксид вуглецю

Обов'язок захищати навколишнє середовище закріплена в Конституції. Тому за компетенції центральних органів влади в області охорони атмосферного повітря віднесені:

- встановлення обмеження на викид забруднюючих речовин в атмосферу;
- визначення порядку державного обліку викидів забруднюючих речовин;
- нормування якості атмосферного повітря;
- визначення порядку видачі спеціальних дозволів (ліцензій) на викиди забруднюючих речовин;
- здійснення державного контролю за охороною атмосферного повітря.

ВИСНОВКИ.

У результаті даної роботи було розроблено модель верифікації інформаційно-вимірювальної системи з обмеженими ресурсами, що представлено Raspberry Pi model 3. Ґрунтуючись на моделі верифікації інформаційних систем і онтологічній моделі для адаптивного web середовища, які надають основні поняття інформаційно-вимірювальних систем і дозволяють їх пов'язати з вимогами до якості і засобами тестування, враховано особливості елементів системи з обмеженими ресурсами для створення відповідної моделі верифікації. також необхідно розширити поняття функціональної одиниці з метою автоматизації процесу тестування не тільки програмного, а й апаратного забезпечення і засобів передачі даних між елементами системи з обмеженими ресурсами.

Практична цінність роботи полягає в тому, що модифікована модель верифікації систем з обмеженими ресурсами, враховує результати верифікації смарт-пристроїв, що входять до інформаційно-вимірювальної системи і описує оптимальний період зчитування лог файлів, що дозволяє забезпечити актуальність і цілісність зберігання результатів контролю систем з елементами з обмеженими ресурсами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hypermedia methods and techniques / E. Knutov, P. De Bra, M. Pechenizkiy // New Review of Hypermedia and Multimedia. – 2009. – Vol. 15, No. 1.– P. 5–38.
2. Corno F. Design-time formal verification for smart environments: an exploratory perspective/ F. Corno, M. Sanaullah, J Ambient //Journal of Ambient Intelligence and Humanized Computing. – 2014. – Vol. 5, Issue 4. – P. 581–599. DOI: 10.1007/s12652- 013-0209-4.
3. Морозов В. П. Сбор и анализ метрик при выполнении проектов программных изделий/ В. П. Морозов, С. Н. Баранов, А. Н. Домарацкий и др. // Программные продукты и системы. – 1998. – № 4. – Режим доступа: – [http://swsys.ru/ index.php?page=article&id=1006](http://swsys.ru/index.php?page=article&id=1006).
4. Вельдер С. Э Верификация автоматных программ. / С. Э. Вельдер, М. А. Лукин, А. А. Шалыто, Б. Р Яминов. - СПб: Наука, 2011. - 244 с.
- 5 Function Point Counting Practices Manual, Release 4.2, IFPUG, 2004. [Електроний ресурс]. – Режим доступа: [http:// www.ifpug.org/](http://www.ifpug.org/).
- 6 Warren G. Raspberry Pi Hardware Reference / G. Warren. – NY: Apress, 2014. – 248 p. DOI:10.1007/978-1-4842-0799-4.
- 7 Віддалений та віртуальний інструментарій в інжинірингу: монографія /[ed. by Karsten Henke]. – Запоріжжя : Дике поле, 2015. – 250 с. ISBN 978-966-2752-74-8.
8. Шитикова Е. В. Динамическая модель передачи данных газотурбинных установок наземного применения / Е. В. Шитикова, Г. В. Табунщик // Наукові праці Донецького національного технічного університету, серія: «Інформатика, кібернетиката обчислювальна техніка». – 2016. – № 1(22). – С. 132-135.
9. Balik M. Generic Ontology-based Model for Adaptive Web Environments / M. Balik, I. Jelinek // IEEE 16th International Conference of Computational Science and Engineering. – 2013. – P. 495–500. DOI: <https://doi.org/10.1109/CSE.2013.80>
10. Айвазян С.А., Прикладна статистика. Основи економетрики: Підручник для вузів: В 2 т. 2-е видання. – Т.1: Теорія імовірностей та прикладна

статистика [Текст] / С.А. Айвазян, В.С. Мхитарян – М.: ЮНИТИ-ДАНА, 2001. – с. 656

11. Кобзарь, А.И. Прикладна математична статистика. Для інженерів та наукових робітників [Текст] / А.И. Кобзарь – М.: ФИЗМАТЛИТ, 2006. – с. 816

12. Chatterjee, Samprit. Handbook of Regression Analysis [Text] / Samprit Chatterjee, Jeffrey S. Somonoff. Wiley, 2012. – 240 p.

13. Нагляд і контроль за дотриманням законодавства про працю і правил з охорони праці [Електронний ресурс]. – Режим доступу: http://lubbook.org/book_240_glava_60_TEMA_11._Nagljad%D1%96_kontrol_z.html – Заг. з екрану

14. Виробниче освітлення [Електронний ресурс]. – Режим доступу: <http://posibnyky.vntu.edu.ua/bjd/45.htm> – Заг. з екрану

15. Гігієна і фізіологія праці [Електронний ресурс]. – Режим доступу: http://intranet.tdmu.edu.ua/data/kafedra/internal/hihiena/lectures_stud/uk/med/lik/ntn/Гігієна_та_екологія/3/06.Гігієна_і_фізіологія_праці.htm – Заг. з екрану

16. Аналіз умов праці на робочому місці користувача ПЕВМ [Електронний ресурс]. – Режим доступу: <http://www.techverious.ru/varwems-721-1.html> – Заг. з екрану

17. Джерела і фактори забруднення навколишнього середовища [Електронний ресурс]. – Режим доступу: http://manyava.ucoz.ua/publ/dzherela_i_faktori_zabrudnennja_navkolishnogo_seredovishha/14-1-0-238 – Заг. з екрану

ВИСНОВКИ

У результаті даної роботи було розроблено модель верифікації даних інформаційно-вимірювальної системи з обмеженими ресурсами, які представляються компонентами на апаратній базі Raspberry Pi model 3.

У першому розділі було проаналізовано існуючі алгоритми верифікації а також викладено теоретичний матеріал, що стосується дослідження.

У другому розділі було викладено матеріал, що стосується саме створення моделі верифікації інформаційно-вимірювальної системи з обмеженими ресурсами.

У третьому розділі було проведено розробку проекту програмного забезпечення, що реалізує створену модель верифікації.

У четвертому розділі було викладено результати розробки програмного забезпечення.

Програмне забезпечення, яке було створено у ході експерименту працює швидко і безвідмовно. Код програмного забезпечення представлено в додатку Б, а результати, отриманні під час експерименту, представлені в додатку В.

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Назва програмного забезпечення – «VSOFT».

Галузь застосування: .

1. Підстави для розробки

Завдання на дипломну роботу магістра, що видано кафедрою ПЗАС та затверджене Наказом ректора НУК № 1037-уч від 26 жовтня 2020 року.

2. Призначення розробки

2.1. Функціональне призначення

Даний програмний продукт призначений для перевірки коректності даних згідно з заданими правилами що надають складові системи, та оперативного інформування відповідальних осіб в разі порушень

2.2. Експлуатаційне призначення

Розроблений додаток повинен використовуватися для верифікації даних від пристроїв

Кінцевий користувач додатку повинен мати базові навички роботи з комп'ютером.

3. Вимоги до програмного продукту

3.1. Вимоги до функціональних характеристик

Надавати можливість задавання системи правил відносно складових системи;

проводити моніторинг та верифікацію;

оперативно реагувати на невідповідності.

3.1.1. Вимоги до складу виконуваних функцій

Додаток повинен надавати користувачу можливість виконання перерахованих нижче функцій:

Надавати можливість задавання системи правил відносно складових системи;

проводити моніторинг та верифікацію;

оперативно реагувати на невідповідності.

3.1.2 Вимоги до організації вхідних та вихідних даних

3.1.3 Вимоги до часових характеристик

Вимоги до часових характеристик не висуваються.

3.2 Вимоги до надійності

Для запобігання відмови програми, внаслідок некоректних дій користувача, програма повинна виконувати перевірку введених даних. У випадку, коли ведені

дані не відповідають вимогам (порожні або містять неприпустимі символи), програма повинна вивести повідомлення про помилку, та запропонувати відредагувати вхідні дані.

3.3 Умови експлуатації

3.3.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, пропонованим до технічних

засобів у частині умов їхньої експлуатації.

3.3.2 Вимоги до видів обслуговування

Програма не вимагає проведення яких-небудь видів обслуговування.

3.3.3 Вимоги до чисельності та кваліфікації персоналу

Вимоги до мінімальної кількості персоналу, необхідного для роботи програми не встановлюють.

Кінцевий користувач програми повинен мати практичні навички роботи із графічним інтерфейсом користувача мобільної операційної системи.

3.4 Вимоги до складу і параметрів технічних засобів

Даний програмний продукт вимагає технічний засіб – персональний комп'ютер, ноутбук, планшет чи мобільний телефон з можливістю виходу в мережу інтернет.

3.5 Вимоги до інформаційної та програмної сумісності

Розроблений програмний продукт орієнтований на роботу в веб-браузерах.

Тому для коректної роботи програми необхідне стабільне функціонування операційної системи та підключення до інтернету.

3.6 Вимоги до маркування та пакування

Поширення даного програмного продукту на фізичних носіях не передбачається.

3.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не пред'являються.

4 Вимоги до програмної документації

Програмна документація повинна містити:

- . технічне завдання;
- . інструкція користувача;
- . опис програмного забезпечення;
- . текст програми;
- . програма та методика випробувань.

5 Техніко-економічні показники

детальне економічне обґрунтування наведено у розділі 5 кваліфікаційної роботи

6 Стадії та етапи розробки

Стадії та етапи розробки представлені в таблиці А.1.

Таблиця А.1 – Стадії та етапи розробки

Стадії розробки	Етапи робіт	Зміст робіт	Терміни виконання
1	2	3	4
Ескізний проект	Розробка ескізного проекту	Попередня розробка структури вхідних та вихідних даних, уточнення методів рішення завдань, загальний опис алгоритму.	З 1.09.20 до 10.09.20
	Затвердження ескізного проекту	Розробка пояснювальної записки. Узгодження і затвердження ескізного проекту.	З 11. 09.20 до 18.09.20
Технічний проект	Розробка технічного проекту	Уточнення структури вхідних та вихідних даних, розробка алгоритму рішення задачі, розробка структури програми.	З 19.09.20 до 8.10.20
	Затвердження технічного проекту	Розробка пояснювальної записки. Узгодження і затвердження технічного проекту.	З 9.10.20 до 26.10.20
	Розробка програми	Програмування та налагодження програми	З 26.10.20 до 5.11.20
	Розробка програмної документації	Розробка програмної документації відповідно до вимог ГОСТ 19.101-77	З 6.11.20 до 7.11.20
	Випробування програми	Розробити, узгодити та затвердити програму і методику випробувань, коригування програми і програмної документації за результатами випробувань.	З 8.11.2020 до 1.12.2020
Робочий проект			

На підставі протоколу проведення випробувань виконавець спільно з замовником підписують акт приймання-здачі програмного забезпечення в експлуатацію.

У разі знаходження помилок під час прийому програмного виробу складається акт про знайдені помилки, який підписується представниками замовника і розробника і затверджується керівниками організації – замовника та організації – розробника. Розробник повинен на протязі не більше ніж 2 тижнів виправити зазначені помилки, і оповістити замовника про повторне проведення перевірки._

ДОДАТОК Б – Інструкція користувача

Користувачем системи є контролер він задає систему правил для верифікації в веб додатку. Після авторизації він потрапляє до системи і бачить головний екран який наведено на рисунку Б1.

Dashboard

Exports

Buildings

Meters

Meteorological Data

Lookup Data

Management

> Dashboard > 5 > Update

Fields with * are required.

OPW Energy Database - Update Dashboard

Organisation: Organisation 3

Building: test356

Status: Public

Save Cancel

Рисунок Б1 – Стартова сторінка

Для надання діапазонів можна вибрати будь-який тип енергії, перелік наведено на рисунку Б2.

Create Meter Conversion Factor

Meter Conversion Factors

ID	User Defined Name	Description	Building	Meter	Resource Unit	Force Recalculation	
252	222	222	test_name	222	All Resource Units	All	
251	new test meter	new test meter	test1	new test meter	Cubic Meter (m3)	No	Q /
250	ooo1	ooo1	test105	ooo1	Kilogram (kg)	No	Q /
248	test3	asdasdaspd	test1	ObviusCH7	kVAR	No	Q /
246	test	oatest	test1	ObviusCH5	kWh	No	Q /
245	4444444	4444444	5050	4444444	Litre (l)	No	Q /
244	21321	21321	5050	21321	MyBTC	No	Q /
242	asdasdghfghjghjghj	asdasdghfghfghfghfgh	222	55555	Pricing (EUR)	No	Q /
239	atest_cv	atest_cv	test1	atest_cv	Pulses	No	Q /
					Tonne (t)	No	Q /

Рисунок Б2 – Типи енергії, облік яких ведеться в системі

Якщо типи лічильників мають різний тип (пульси, кіловати тощо) для коректної верифікації треба занести фактори конверсії наведено на рисунку БЗ.

The screenshot shows a web application interface for managing meter conversion factors. On the left is a dark sidebar with navigation links: Dashboard, Exports, Buildings, Meters (highlighted in red), Meters List, Meters Tree, Meter Conversion Factors, Suppliers, Units, Meteorological Data, Lookup Data, and Management. The main content area is titled 'Meter Conversion Factors - 222'. It features three buttons: 'Meter Conversion Factors', 'Update Meter Conversion Factor', and 'Create Meter Conversion Factor'. Below these are input fields for: ID (252), Description (222), Energy Type (Gas), Resource Unit (kWh), User Defined Name (222), Building (11601_name), Meter (222), and Force Recalculation (No). A 'Create Meter Conversion Factor Detail' button is also present. At the bottom, a table titled 'Meter Conversion Factors Detail' has columns for ID, Factor, Start Datetime, and End Datetime. The first row shows ID 259, Factor 5.0000000000, Start Datetime 01/01/2020 00:00, and an empty End Datetime field.

Риунок БЗ – Типи лічильників

Logger Name	Severity	Activity	Created	
CPW000080	WARNING	Alert for logger CPW000080: Channels 1, 2, 3 are not setup but received value.	25/11/2020 14:27	Q
001EC60542E8	WARNING	Alert for logger 001EC60542E8: Channels 1, 2, 3, 6, 8, 9, 10, 11, 12 are not setup but received value.	19/11/2020 22:29	Q
001EC60542E8	INFO	HTTP FILE IMPORT: File import finish successfully at 2020-11-19 21:20:39	19/11/2020 22:28	Q
001EC60542E8	WARNING	Alert for logger 001EC60542E8: Channels 1, 2, 3, 6, 8, 9, 10, 11, 12 are not setup but received value.	19/11/2020 22:05	Q
001EC60542E8	INFO	HTTP FILE IMPORT: File import finish successfully at 2020-11-19 21:05:56	19/11/2020 22:05	Q
001EC60542E8	WARNING	Alert for logger 001EC60542E8: Channels 1, 2, 3, 6, 8, 9, 10, 11, 12 are not setup but received value.	19/11/2020 21:51	Q
001EC60542E8	INFO	HTTP FILE IMPORT: File import finish successfully at 2020-11-19 20:51:24	19/11/2020 21:51	Q

Рисунок Б4 – Звіти з обліку репортів

Подробиці лог файлів звітів з проблемами верифікації можна подивитися в розділі Логгер Репорте.

User Profile Settings

Profile Settings

Username: test

First name: testovich

Last name: testov

User phone:

Password:

Repeat password:

Receive Alert Emails: ☐

Choose File: No file chosen

Рисунок Б5 – Налаштування отримання повідомлень про проблеми з логером в профілі користувача-контролера

Profile Settings

Working days *

Working hours *

Receive SMS *

Send alerts only on working hours *

Date format *

Update

Рисунок Б6 – Вибір засобу та часу надсилання повідомлень про валідаційні проблеми

Logger Create

Logger Ref *

Building Name *

Logger Type *

Communication Method *

Data Provider *

Location Details *

Notes

Logger Name *

Logger Phone *

Logger Email *

Link by

Subject *

Tracking Code *

Logger URL *

Communication Frequency *

Logger password *

Alert *

Is active *

Generate

Activation Windows

Back Save

Рисунок Б7 – Створення нового логера

Communication Frequency * 15 min

Logger password Generate

Alert ☒ 1 days

Is active ☒

Рисунок Б8 – Редагування часового обсягу логу верифікації



Рисунок Б9 – Результати верифікації апаратного забезпечення (логерів) на дашборді

Building Name *

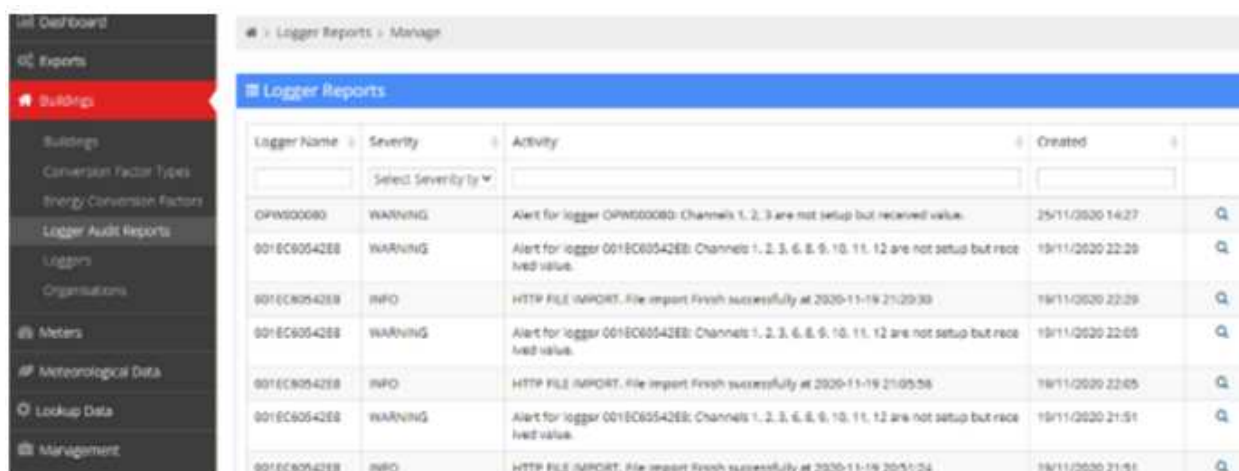
Start date/time *

Read Source ☐ Meter Value (As Read)

☒ Utility Bill or Calculated Value

End date/time

Рисунок Б10 – Визначення часу запуску верифікації



The screenshot shows a web application interface with a sidebar on the left containing navigation links: Dashboard, Exports, Buildings (highlighted), Buildings, Conversion Factor Types, Energy Conversion Factors, Logger Audit Reports, Loggers, Organisations, Meters, Meteorological Data, Lookup Data, and Management. The main content area is titled 'Logger Reports > Manage' and displays a table of 'Logger Reports'.

Logger Name	Severity	Activity	Created
OPW00080	WARNING	Alert for logger OPW00080: Channels 1, 2, 3 are not setup but received value.	25/11/2020 14:27
001EC60542EB	WARNING	Alert for logger 001EC60542EB: Channels 1, 2, 3, 6, 8, 9, 10, 11, 12 are not setup but received value.	19/11/2020 22:29
001EC60542EB	INFO	HTTP FILE IMPORT, File import Finish successfully at 2020-11-19 21:09:30	19/11/2020 22:29
001EC60542EB	WARNING	Alert for logger 001EC60542EB: Channels 1, 2, 3, 6, 8, 9, 10, 11, 12 are not setup but received value.	19/11/2020 22:05
001EC60542EB	INFO	HTTP FILE IMPORT, File import Finish successfully at 2020-11-19 21:05:58	19/11/2020 22:05
001EC60542EB	WARNING	Alert for logger 001EC60542EB: Channels 1, 2, 3, 6, 8, 9, 10, 11, 12 are not setup but received value.	19/11/2020 21:51
001EC60542EB	INFO	HTTP FILE IMPORT, File import Finish successfully at 2020-11-19 20:51:24	19/11/2020 21:51

Рисунок Б11 – Звіти з результатів верифікації інформаційно виміральної системи

ДОДАТОК В – ТЕКСТ ПРОГРАМИ

```

<template>
  <div id="app">
    <div v-if="! auth">
      <div class="container auth">
        <div class="w-25">
          <b-form-group>
            <b-form-input type="text" placeholder="login" v-model="login"/>
          </b-form-group>
          <b-form-group>
            <b-form-input type="password" placeholder="password" v-
model="password"/>
          </b-form-group>
          <br>
          <b-alert v-model="error.show" variant="danger" dismissible>
            {{error.message}}
          </b-alert>
          <br>
          <b-button class="btn" variant="success" @click="getAuth">Sign
In</b-button>
        </div>
      </div>
    </div>
    <div v-else>
      <div class="row auth" v-if="! building">
        <div class="form-group col-sm-2">
          <label for="building">Please select building: </label>
          <select id="building" class="form-control" v-model="building">

```

```

        <option selected v-for="(value, label) in options.buildings"
:value="value" :key="value">
            {{ label }}
        </option>
    </select>
</div>
</div>
<div v-if="building">

    <weather></weather>
    <div class="row charts">
        <div class="col-sm-3 cusum">
            <cuSum
                :building="building"
                :energy_type="options.energy_type"
                :meters="options.meters[building]"
                :stepsLabels="options.cusumStepsLabels"
                :degdyLabels="options.degdyLabels"
            ></cuSum>
        </div>
        <div class="col-sm-3 heatmap">
            <heatMap
                :building="building"
                :energy_type="options.energy_type"
                :meters="options.meters[building]"
                :stepsLabels="options.heatmapStepsLabels"
            ></heatMap>
        </div>
        <div class="col-sm-6 consumption">
            <consumption

```

```

        :building="building"
        :energy_type="options.energy_type"
        :meters="options.meters[building]"
        :stepsLabels="options.consumptionStepsLabels"
      ></consumption>
    </div>
  </div>
</div>
<div class="text-center spinner">
  <b-spinner style="width: 5rem; height: 5rem;" type="grow" large v-
if="loading"></b-spinner>
</div>
</div>
</template>

```

```

<script>
  export default {
    name: 'App',
    data: () => ({
      login: "",
      password: "",
      options: null,
      loading: false,
      auth: false,
      building: null,
      meter: null,
      error: {
        show: false,
        message: ""
      }
    })
  }

```

```

    },
  }),
  methods: {
    async getAuth() {
      this.axios.defaults.headers.common['PHP_AUTH_USER'] = this.login;
      this.axios.defaults.headers.common['PHP_AUTH_PW'] =
this.password;
      this.axios.defaults.headers.common['LOGIN'] = this.login;
      this.axios.defaults.headers.common['PASSWORD'] = this.password;
      this.axios.defaults.baseURL =
process.env.VUE_APP_URL_ENDPOINT;
      this.axios.defaults.timeout = 1000;
      this.axios.interceptors.request.use((config) => {
        // Do something before request is sent
        this.loading = true;
        return config;
      }, (error) => {
        // Do something with request error
        return Promise.reject(error);
      });
      this.axios.interceptors.response.use((response) => {
        // Any status code that lie within the range of 2xx cause this function
to trigger

        // Do something with response data
        this.loading = false;
        return response;
      }, (error) => {
        // Any status codes that falls outside the range of 2xx cause this
function to trigger

        // Do something with response error

```

```

        this.loading = false;
        return Promise.reject(error);
    });
    try {
        let options = await this.axios.get('/options');
        this.options = options.data;
        this.auth = true;
        this.error.show = false;
    } catch (e) {
        if (e.response) {
            console.log(e.response.data);
            this.error.message = e.response.data.warning;
            this.error.show = true;
        }
    }

    },
    },
}
</script>

<style>
    .charts {
        justify-content: center;
    }

    .cusum {
        margin-right: 2rem;
        margin-left: 0rem;
    }

```



```
padding-left: 0;
}
.consumption {
margin-left: 0rem;
padding-left: 0;
}
```

```
.heatmap{
margin-right: 0;
padding-right: 0;
}
```

```
.auth {
height: 100vh;
text-align: center;
display: flex;
justify-content: center;
align-items: center;
}
```

```
.spinner {
position: absolute;
z-index: 999;
top: 50%;
left: 50%;
}
```

```
.btn-go {
```

```

        width: 100%;
    }

    .settings {
        padding: 1rem
    }

    .range-picker{
        min-width: 400px;
    }

    #app{
        background-color: #F0F0F0;
    }
</style>
<template>
    <div class="container settings">
        <b-alert v-model="error.show" variant="danger" dismissible>
            {{error.message}}
        </b-alert>

        <div class="row" v-show="settings">

            <div class="form-group col-sm-12 text-center">
                <label for="date-time-picker">Date and time range: </label><br>
                <date-time-range-picker
                    @updatePicker="getDateTimeRangePickerData($event)"
                    id="date-time-picker"

```

```

        :date-start="dateStart"
        :date-end="dateEnd"
    >
</date-time-range-picker>
</div>

```

```

<div class="form-group col-sm-6">
    <label for="step">Step</label>
    <select id="step" class="form-control" v-model="params.step">
        <option v-for="(label, value) in stepsLabels" :value="value"
:key="value">
            {{ label }}
        </option>
    </select>
</div>

```

```

<div class="form-group col-sm-6" v-if="! params.energy_type_flag">
    <label for="meter">Meter</label>
    <select id="meter" class="form-control" v-model="params.meter"
multiple>
        <option v-for="(value, label) in meters" :value="value"
:key="value">
            {{ label }}
        </option>
    </select>
</div>

```

```

<div class="form-group col-sm-6 d-block">
    <div class="col-sm-12 m-b-30">

```

```

<div class="custom-control custom-checkbox">
  <input type="checkbox" class="custom-control-input" value="1"
name="dst-status" id="dst"
      v-model="params.dst">
  <label class="custom-control-label v-a-m" for="dst">DST</label>
</div>
</div>
<div class="col-sm-12">
  <div class="custom-control custom-checkbox">
    <input type="checkbox" class="custom-control-input" value="1"
name="energy-status"
        id="meter_energy_type"                                v-
model="params.energy_type_flag"/>
    <label class="custom-control-label v-a-m"
for="meter_energy_type">Meter/Energy type</label>
  </div>
</div>
</div>

<div class="form-group col-sm-6">
  <label for="chart-type">Chart type</label>
  <select id="chart-type" class="form-control" v-model="chartType">
    <option v-for="(label, value) in chartTypes" :value="value"
:key="value">
      {{ label }}
    </option>
  </select>
</div>

```

```

        <div class="form-group col-sm-12 align-self-lg-end">
            <button          class="btn          btn-primary          btn-go"
@click="getData">Go</button>
        </div>

    </div>

    <div class="row">
        <div class="chart-container">
            <line-chart  v-if="chartType  ===  'line'"  :chart-data="chartData"
:width="900"/>
            <bar-chart  v-else-if="chartType  ===  'bar'"  :chart-data="chartData"
:width="900"/>
        </div>
    </div>

    <div class="text-center">
        <b-button-group>
            <b-button variant="outline-primary" @click="getData">
                <b-icon icon="arrow-repeat"></b-icon>
            </b-button>
            <b-button variant="outline-primary" @click="settings = !settings">
                <b-icon icon="pencil"></b-icon>
            </b-button>
        </b-button-group>
    </div>

</div>

```

</template>

<script>

```
import moment from 'moment'
```

```
export default {  
  name: "ConsumptionComponent",  
  props: {  
    building: {  
      type: Number,  
      required: true  
    },  
    meters: {  
      type: Object,  
      required: false  
    },  
    energy_type: {  
      type: Object,  
      required: false  
    },  
    stepsLabels: {  
      type: Object,  
      required: true  
    },  
  },  
  data: () => ({  
    chartTypes: {  
      line: 'line',  
      bar: 'bar',
```

```

    },
    chartType: null,
    dateStart: moment().subtract(1, 'day').startOf('day'),
    dateEnd: moment().endOf('day'),
    params: {
        meter: null,
        energy_type_flag: true,
        energy_type: null,
        step: null,
        dst: false,
        startDate: null,
        endDate: null,
    },
    chartData: null,
    error: {
        show: false,
        message: ""
    },
    settings: false
  )),
  mounted() {
    this.params.energy_type = Object.values(this.energy_type)[0];
    this.params.step = Object.keys(this.stepsLabels)[0];
    this.chartType = this.chartTypes.line;
    this.updateData();
  },
  methods: {
    getDateTimeRangePickerData($event) {
      this.params.startDate = $event.startDate;
      this.params.endDate = $event.endDate;
    }
  }
}

```

```

    },
    async getData() {
        var data = {
            building_id: this.building,
            energy_type_flag: this.params.energy_type_flag,
            step: this.params.step,
            dst: this.params.dst,
            from: this.params.startDate,
            to: this.params.endDate,
        };

        if (this.params.energy_type_flag) {
            data.energy_id = this.params.energy_type;
        } else {
            data.meter_id = this.params.meter;
        }

        try {
            var response = await this.axios.post('/chart/consumption', data);
            this.chartData = fillDataCollection(response.data, this.chartType);

            this.error.show = false;
        } catch (e) {
            if (e.response) {
                console.log(e.response.data.warning);
                this.error.message = e.response.data.warning;
                this.error.show = true;
            }
        }
    }

```



```

    },
    updateData() {
        this.getData();
        setInterval(() => {
            this.getData();
        }, (20 * 60 * 1000));
    },
},
}

```

```

function fillDataCollection(data, chartType) {
    let chartData = prepareConsumptionData(data, chartType);

    return {
        labels: data.dates,
        datasets: chartData
    }
}

```

```

function prepareConsumptionData(data, chartType) {
    let consumptionData = [];
    var colors = [
        '#f87979',
        '#A5CC82',
        '#BECEDA',
        '#FFFFCC',
        '#AEC123',
        '#EDA321',
        '#456AFD',
    ];
}

```

```

    '#112233',
    '#098543',
    '#39AC11',
  ];

```

```

Array.from(data.values).forEach(function (entry, index) {

```

```

    let backgroundColor = null;
    if (chartType === 'line') {
      backgroundColor = 'transparent';
    } else if (chartType === 'bar') {
      backgroundColor = colors[index] ? colors[index] : null;
    }

```

```

    consumptionData.push({
      label: entry.label,
      lineTension: 0,
      borderWidth: 2,
      borderColor: colors[index] ? colors[index] : null,
      pointBackgroundColor: colors[index] ? colors[index] : null,
      backgroundColor: backgroundColor,
      data: entry.data
    })
  });

```

```

    return consumptionData;
  }

```

```

</script>

```

<style>

</style>

ДОДАТОК Г –ТЕСТУВАННЯ

Експеримент № 1

Розмір Масиву = 10000

| Кількість | Час | Швидкість (e\с) |
|-----------|---------------------|-----------------|
| 1 | 0.44482994079589844 | 22480.501159854 |
| 2 | 0.4626309871673584 | 21615.49977711 |

Час обробки програми збільшився на +4%

Експеримент № 2

Розмір Масиву = 20000

| Кількість | Час | Швидкість (e\с) |
|-----------|--------------------|-----------------|
| 1 | 1.9904301166534424 | 10048.079474213 |
| 2 | 0.8577709197998047 | 23316.248590787 |

Час обробки програми зменшився на 56.91%

Експеримент № 3

Розмір Масиву = 30000

| Кількість | Час | Швидкість (e\с) |
|-----------|--------------------|-----------------|
| 1 | 1.4945099353790283 | 20073.469764115 |
| 2 | 1.2666800022125244 | 23683.961180092 |

Час обробки програми зменшився на 15.24%

Експеримент № 4

Розмір Масиву = 40000

| Кількість | Час | Швидкість (e\c) |
|-----------|--------------------|-----------------|
| 1 | 2.06109619140625 | 19407.148568213 |
| 2 | 1.6610851287841797 | 24080.644216758 |

Час обробки програми зменшився на 19.41%

Експеримент № 5

Розмір Масиву = 50000

| Кількість | Час | Швидкість (e\c) |
|-----------|--------------------|-----------------|
| 1 | 2.6265389919281006 | 19036.458302603 |
| 2 | 2.0819249153137207 | 24016.235951749 |

Час обробки програми зменшився на 20.74%

Експеримент № 6

Розмір Масиву = 60000

| Кількість | Час | Швидкість (e\c) |
|-----------|--------------------|-----------------|
| 1 | 3.1747350692749023 | 18899.214797695 |
| 2 | 2.552399158477783 | 23507.295009368 |

Час обробки програми зменшився на 19.60%

Експеримент № 7

Розмір Масиву = 70000

| Кількість | Час | Швидкість (e\c) |
|-----------|-------------------|-----------------|
| 1 | 3.770120143890381 | 18567.047555086 |
| 2 | 2.943565845489502 | 23780.680872916 |

Час обробки програми зменшився на 21.92%

Експеримент № 8

Розмір Масиву = 80000

| Кількість | Час | Швидкість (e\c) |
|-----------|-------------------|-----------------|
| 1 | 4.407039165496826 | 18152.777181181 |
| 2 | 3.378964900970459 | 23675.889612533 |

Час обробки програми зменшився на 23.33%

Експеримент № 9

Розмір Масиву = 90000

| Кількість | Час | Швидкість (e\c) |
|-----------|--------------------|-----------------|
| 1 | 5.038016080856323 | 17864.174817144 |
| 2 | 3.8176209926605225 | 23574.891319235 |

Час обробки програми зменшився на 24.22%

Експеримент № 10

Розмір Масиву = 100000

| Кількість | Час | Швидкість (e\с) |
|-----------|-------------------|-----------------|
| 1 | 5.536394119262695 | 18062.297922771 |
| 2 | 4.260785102844238 | 23469.852993348 |

Час обробки програми зменшився на 23.04%

Експеримент № 11

Розмір Масиву = 200000

| Кількість | Час | Швидкість (e\с) |
|-----------|--------------------|-----------------|
| 1 | 11.953681945800781 | 16731.246565437 |
| 2 | 8.755303859710693 | 22843.296269858 |

Час обробки програми зменшився на 26.76%

Експеримент № 12

Розмір Масиву = 300000

| Кількість | Час | Швидкість (e\с) |
|-----------|--------------------|-----------------|
| 1 | 18.633754014968872 | 16099.815408049 |
| 2 | 13.362910985946655 | 22450.198187768 |

Час обробки програми зменшився на 28.29%

Експеримент № 13

Розмір Масиву = 400000

| Кількість | Час | Швидкість (e\с) |
|-----------|--------------------|-----------------|
| 1 | 25.380781888961792 | 15759.955770865 |
| 2 | 18.095077991485596 | 22105.458743434 |

Час обробки програми зменшився на 28.71%

Експеримент № 14

Розмір Масиву = 500000

| Кількість | Час | Швидкість (e\с) |
|-----------|--------------------|-----------------|
| 1 | 32.15199017524719 | 15551.136874411 |
| 2 | 22.656049013137817 | 22069.161295955 |

Час обробки програми зменшився на 29.53%

Експеримент № 15

Розмір Масиву = 600000

| Кількість | Час | Швидкість (e\с) |
|-----------|-------------------|-----------------|
| 1 | 39.73838186264038 | 15098.752688873 |
| 2 | 27.88145089149475 | 21519.683546419 |

Час обробки програми зменшився на 29.84%

Експеримент № 16

Розмір Масиву = 700000

| Кількість | Час | Швидкість (e\с) |
|-----------|--------------------|-----------------|
| 1 | 47.026036977767944 | 14885.370849577 |
| 2 | 32.90591597557068 | 21272.770541312 |

Час обробки програми зменшився на 30.03%

Час обробки програми зменшився на 37.33%

ДОДАТОК Д – ОПИС ПРОГРАМИ

Загальні відомості

1.1 Позначення і найменування програми

Найменування: Програмне забезпечення для верифікації складових інформаційно-вимірювальних систем

Позначення: Верифікаційна інформаційно-вимірювальна система.

1.2 Програмне забезпечення, необхідне для функціонування програми

Браузер

1.3 Мови програмування

Програмне забезпечення в для верифікації інформаційно-вимірювальної системи є комплексним і реалізовано на різних мовах вебдодаток для надання правил реалізований на мові високого рівня PHP, інформаційна система моніторингу що входить до її складу реалізована на Vue.js, а на мікрокомп'ютерах для розподілу ресурсів використовуються python скрипти.

2 Функціональне призначення

2.1 Класи вирішуваних завдань і призначення програми

Програмне забезпечення

створене для верифікації даних що надходять від складових системи;
оперативного інформування відповідних осіб;
моніторингу стані системи.

2.2 Функціональні обмеження

3 Опис логічної структури

3.1 Загальна структура

Dashboard реалізований на VUE JS

Система задання правил верифікації на Laravel Nova (PHP)

ПЗ налаштувань смарт реєстраторів на Python

ДОДАТОК Є – ПРОГРАМА І МЕТОДИКА ВИРОБУВАНЬ

1 Об'єкт випробувань

1.1 Найменування

Програмне забезпечення верифікації даних.

1.2 Галузь застосування енергетика

1.3 Умовне позначення

Програмне забезпечення верифікації даних

2 Ціль випробувань

Програмне забезпечення верифікації даних, що проводяться по даній програмі і методиці є визначення функціональної працездатності ПЗ на етапі проведення випробувань.

Програма випробувань повинна засвідчити працездатність ПЗ в відповідності з функціональним призначенням та вимогам визначених в програмному документі «Технічне завдання».

3 Вимоги до програми

Програмне забезпечення повинно відповідати наступним вимогам

- перевіряти правильність даних що надходять від логера;
- перевіряти, що відповідь від логера не перевищує заданий діапазон;
- надсилати повідомлення про невдалу верифікацію користувачу контролеру.

4 Вимоги до програмної документації

4.1 Перелік програмної документації, що пред'являється на випробування

Програмна документація повинна містити:

- технічне завдання;
- інструкцію користувача;
- опис програми;
- тексти програмних модулів;
- програму та методику випробувань.

4.2 Спеціальні вимоги

Спеціальні вимоги до програмної документації не пред'являються.