

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено
Завідувач кафедри

О.В.Коваль
(ініціали, прізвище)

_____ (підпис)

“ ____ ” _____ 2019 р.

ДИПЛОМНА РОБОТА

на здобуття ступеня бакалавра

з напрямку підготовки
6.050103 “Програмна інженерія”

на тему: Web-інтерфейс стенду моделювання віддаленого моніторингу та управління технологічним процесом

Виконав: студент 4 -курсу, групи ТВз-51 -

Вовчок Володимир Іванович

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник доцент, к.т.н. Ковальчук Артем Михайлович

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

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

Студент _____

(підпис)

Київ – 2019

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050103 “Програмна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.В. Коваль
(підпис)

” ” _____ 2019 р.

ЗАВДАННЯ

на дипломну роботу студенту

Вовчку Володимирі Івановичу

(прізвище, ім'я, по батькові)

1. Тема роботи “Web-інтерфейс стенду моделювання віддаленого моніторингу та управління технологічним процесом”

керівник роботи доцент, к.т.н. Ковальчук Артем Михайлович
(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” ___ ” _____ 201__ р.
№ _____.

2. Строк подання студентом роботи ___ _____ 201__ р.

3. Вихідні дані до роботи стенд, персональний комп'ютер під керуванням операційної системи Microsoft Windows або інший девайс, мова програмування JAVA.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати існуючі програмні рішення та можливі засоби реалізації взаємодії, обґрунтувати обрані програмні застосунки та шляхи розробки програмних додатків, розробити програмне забезпечення, розробити користувацький інтерфейс, зробити висновки за результатами роботи

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових креслень)

1. Загальна структура віддаленого доступу з використанням мережі Інтернет;

2. Інтерфейс користувача; 3. Типова структура web-додатки; 4. Стенд моделювання

віддаленого моніторингу та управління технологічним процесом; 5.Стенд моделювання віддаленого моніторингу та управління технологічним процесом (мікрокомп'ютер, мікроконтроллер, модулі); 6– 8. Стенд моделювання; 9. Опис стенду моделювання віддаленого моніторингу та управління технологічним процесом; 10.Інтерфейс користувача (апробація)

6. Публікації: _____

Дата видачі завдання ” ____ ” _____ 201__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі		
2.	Розробка архітектури та загальної структури системи		
3.	Розробка структур окремих підсистем		
4.	Підготовка матеріалів		
5.	Програмна реалізація системи		
6.	Захист програмного продукту		
7.	Оформлення пояснювальної записки		
8.	Передзахист		
9.	Захист		

Студент

_____ (підпис)

Вовчок В. І. _____ Д
(прізвище та ініціали)

Керівник роботи

_____ (підпис)

Ковальчук А. М. _____ а
(прізвище та ініціали)

АНОТАЦІЯ

Дипломна робота присвячена розробці Web-інтерфейсу стенду моделювання віддаленого моніторингу та управління технологічним процесом. На сучасному етапі розвитку технічних та інформаційних систем існує необхідність розвитку технологій віддаленого моніторингу та управління технічними об'єктами з використанням мережі Internet. З огляду на стрімкий розвиток технологій ІОТ та ІІОТ, Industry 4.0, ця тема є дуже актуальною на даний момент, тому що віддалений моніторинг та управління дозволяють будь-якому користувачу наглядати за сервером без фізичного доступу та допомоги системного адміністратора.

Метою роботи є розробка Web-інтерфейсу стенду моделювання віддаленого моніторингу та управління технологічним процесом.

В ході виконання дипломної роботи була побудована робоча модель системи, реаліовано Web-інтерфейс, а також тестовий стенд з мікрокомп'ютером, мікроконтролером та модулями.

Записка містить 53 сторінок, 10 рисунків, 6 додатків і 29 посилань.

Ключові слова: Java 8EE, HTML, CSS, EJB, JSSC, DTS, Vaadin Framework, SEFL SDK (Configuration, UART), JUnit, JetBrains IntelliJ IDEA, Apache Tomcat, Віджет, Гаджет, Девайс.

ABSTRACT

Thesis is devoted to the development of the Web-interface of the stand of simulation of remote monitoring and control of the technological process. At the present stage of development of technical and information systems, there is a need for the development of technologies for remote monitoring and management of technical objects using the Internet. Given the rapid development of IOT and ITOT, Industry 4.0, this topic is very topical at the moment, because remote monitoring and management allow any user to monitor the server without physical access and help the system administrator.

The purpose of the work is to develop a web-interface of the stand of simulation of remote monitoring and control of the technological process.

During the execution of the thesis, a working model of the system was constructed, the Web-interface was implemented, as well as a test bench with a microcomputer, a microcontroller and modules.

The note contains 53 pages, 10 figures, 6 attachments and 29 links.

Keywords: Java 8EE, HTML, CSS, EJB, JSSC, DTS, Vaadin Framework, SEFL SDK (Configuration, UART), JUnit, JetBrains, IntelliJ IDEA, Apache Tomcat, Widget, Gadget, Devays.

АННОТАЦИЯ

Дипломная работа посвящена разработке Web-интерфейса стенда моделирования удаленного мониторинга и управления технологическим процессом. На современном этапе развития технических и информационных систем существует необходимость развития технологий удаленного мониторинга и управления техническими объектами с использованием сети Internet. Учитывая стремительное развитие технологий ИВТ и ИИот, Industry 4.0, эта тема является очень актуальной на данный момент, так как удаленный мониторинг и управление позволяют любому пользователю следить за сервером без физического доступа и помощи системного администратора.

Целью работы является разработка Web-интерфейса стенда моделирования удаленного мониторинга и управления технологическим процессом.

В ходе выполнения дипломной работы была построена рабочая модель системы, реализовано Web-интерфейс, а также тестовый стенд с микрокомпьютером, микроконтроллером и модулями.

Записка содержит 53 страниц, 10 рисунков, 6 приложений и 29 ссылок.

Ключевые слова: Java 8EE, HTML, CSS, EJB, JSSC, DTS, Vaadin Framework, SEFL SDK (Configuration, UART), JUnit, JetBrains IntelliJ IDEA, Apache Tomcat, Виджет, Гаджет, Девайс.

ЗМІСТ

Перелік умовних позначень, скорочень і термінів	7
ВСТУП.....	8
1 ПОСТАНОВКА ЗАДАЧІ.....	11
2 АНАЛІЗ ПРОБЛЕМ РОЗРОБКИ WEB-ІНТЕРФЕЙСУ СТЕНДУ МОДЕЛЮВАННЯ ВІДДАЛЕНОГО МОНІТОРИНГУ ТА УПРАВЛІННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ.....	13
2.1 Складові упровадження моніторингу технологічних процесів	14
2.2 Віддалений моніторинг та керування технологічними об'єктами за допомогою web-технологій	17
2.3 Існуюча реалізація віддаленого моніторингу та управління технологічним процесом.....	22
Висновки до розділу 2	24
3 WEB-ІНТЕРФЕЙС	26
3.1 Web-інтерфейс	26
3.2 Інтерфейс користувача	28
Висновки до розділу 3	31
4 ОБГРУНТУВАННЯ ВИБРАНИХ ЗАСОБІВ РОЗРОБКИ	32
4.1 Обґрунтування вибору мови програмування та фреймворку	32
4.2 Обґрунтування вибору середовища розробки	36
4.3 Структура web-додатку	40
Висновки до розділу 4	42
5 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	43
Висновки до розділу 5	48
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТКИ.....	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Java 8EE — Java Platform Enterprise Edition 8

HTML — HyperText Markup Language

CSS — Cascading Style Sheets

DOM — Document Object Model

EJB — Enterprise Java Beans.

JSSC — Java Simple Serial Connector

DTS — Data Transformation Services

SEFL SDK — Configuration, UART; Software development kit

Vaadin Framework — фреймворк для створення web-додатків

JUnit — фреймворк автоматичного тестування коду

JetBrains Intelij IDEA — комерційне інтегроване середовище розробки для різних мов програмування

Apache Tomcat — контейнер сервлетів

ІТ — Інформаційні технології

ОС — Операційна система

ПЗ — Програмне забезпечення

ПК — Персональний комп'ютер

Віджет — Графічний засіб відображення інформації на екрані персонального комп'ютеру чи смартфона

Гаджет — Обчислювальний пристрій. Частіше за все смартфон чи планшет

Девайс — пристрій, пристосування, яке використовується в багатьох областях, призначене для виконання приватної, спеціальної задачі

ВСТУП

Світ технологій дуже швидко розвивається. Цей розвиток іде безперервно і динамічно. Кожен рік приносить нам нові концепції та їх реалізації, а існуючі більш розвиваються, розширюють свої можливості, функції та покращують характеристики. Мета прогресу, який відбувається не нова – покращити умови життя людини, робити його більш комфортнішим і продуктивним. Головною причиною глобальних змін в характеристиках та швидких темпах зростання ефективності впровадження технологій в життя людини є широке поширення Internet, а також значне збільшення швидкості передачі даних.

Одним із варіантів покращення ефективності життя та роботи є правильна та ефективна робота його технологічних об'єктів, а також підтримка їх оптимальних режимів роботи. Однією з головних задач, які з'являються при розробці і впровадженні автоматизованих систем керування технологічними об'єктами, є забезпечення безперервного моніторингу, контролю та оперативного віддаленого керування технологічним процесом. Ця задача стає найбільш важливою тоді, коли об'єкти, з якими працює система можуть бути на великій відстані територіально один від одного, або у важкодоступних або небезпечних для людини місцях.

На теперішньому етапі розвитку інформаційних та технічних систем виникла необхідність в процесі поступових змін технологій віддаленого керування технічними об'єктами з використанням мережі Internet. У роботі розроблена загальна структура стенду моніторингу та управління з доступом через Internet для віддаленого виконання. Для втілення в життя функцій, які контролюють режими роботи та керування технічними об'єктами можуть застосовуватись як стандартні рішення, так і нетрадиційні експериментальні розробки.

Аналізуючи теоретичну інформаційну базу щодо комп'ютерного моделювання, зрозуміло, що воно є вигідним інструментом для розробника, тому що дає можливість завчасно та з найменшим марнуванням часових і матеріальних ресурсів передбачити поведінку системи, яка розроблюється.

Висловлюючись про новизну віддаленого моніторингу та управління технологічним процесом, потрібно згадати скоріше не про інноваційність, а про перехід від кількості в якість. Зараз практично будь-який пристрій з будь-якою платформою може працювати завдяки синхронізації через віддалені сервіси.

У даній роботі висвітлено практичну значущість моделювання віддаленого моніторингу та управління технологічним процесом, яке надає можливість користувачу спростити роботу та розв'язати безліч завдань, виходячи із заданих умов середовища таких, як вибір оптимальних параметрів розроблюваної системи.

При виконанні практичної частини дипломної роботи було розглянуто декілька існуючих технологій створення взаємодії та обміну даних між додатками. Серед найпоширеніших мов web-програмування було обрано та використано в розробці: HTML, CSS, Java. Також перевагою цієї системи є не тільки широка сфера застосування, а й те, що вона працює на будь яких операційних системах.

Мета виконання дипломної роботи полягає у є розробці Web-інтерфейсу стенду моделювання віддаленого моніторингу та управління технологічним процесом. А також надання широкого доступу до втілення в життя такої системи, яку можна буде вільно видозмінювати і переробляти будь-якому користувачу при бажанні.

Для досягнення поставленої мети необхідно виконати такі завдання:

- визначити завдання для системи моніторингу;
- сформулювати вимоги до розроблення системи моніторингу;
- розробити структуру та обрати засоби впровадження системи моніторингу технологічних процесів;
- збір технологічних даних у реальному часі;
- збереження та підготовче опрацювання даних;
- аналітичне опрацювання даних;
- формування сумов для управління виконавчими механізмами;
- формування звітів про стан розробки системи.

У першому розділі записки сформульовано постановку задачі й розглянуто проблеми щодо специфіки розробки web-інтерфейсу, у другому роділі

проаналізовані проблеми розробки web-інтерфейсу, третій розділ деталізує послідовність розробки інтерфейсу, у четвертому розділі описано засоби розробки; у п'ятому розділі розглянуто програмну реалізацію.

1 ПОСТАНОВКА ЗАДАЧІ

Існує декілька типів взаємодій, одною з яких є людино-машинна взаємодія.

Людино-машинна взаємодія (англійською — HCI) — це вивчення, планування, розробка та розвиток взаємодії між користувачами і комп'ютерними машинами та системами. Цей цикл існує для розробки вдосконалення методів вдосконалення, оцінки та впровадження інтерактивних комп'ютерних систем, які призначені для того, щоб їх могла використовувати людина. Важливою стороною людино-комп'ютерної взаємодії є гарантування виконання запитів користувачів.

Одним із головних апитів користувачів є якісна розробка інтерфейсу. Адже інтерфейси, які мають неякісну розробку можуть стати причиною багатьох непередбачених проблем.

Основним завданням, які ставить перед собою людино-комп'ютерна взаємодія, є поліпшення взаємодії між людиною і машиною, тобто є потреба робити комп'ютери більш зручними і адаптованими до потреб користувачів.

Щодо завдання людино-комп'ютерної взаємодії, яке є довгостроковим це зменшення бар'єру між людиною і комп'ютером.

Міжмашинна взаємодія (англійською — IPC) — це ще один вид взаємодії, який за участі мережевого зв'язку, сприяє стрімкому й дієвому обміну інформацією між різними приладами. Дана технологія поєднує об'єкти, які знаходяться на відстані один від одного, для подальшої автоматизації процесів. Бездротові системи реалізує моніторинг різних датчиків, сенсорів та іншого обладнання між собою. Ця «співпраця» допомагає оптимально покращити процес управління. Міжмашинна взаємодія — виконується через спеціальну абстракцію — сокет. Залежно від рівня його використання, є такі засоби міжмашинної взаємодії процесів:

- пряме використання сокетів — технологія, що потребує програмування на низькому рівні і реалізації протоколу передачі даних;
- віддалений виклик процедур — RPC-технологія, гарантує взаємодію між процесами яка схожа до виклику функцій, при якому дані передаються в один бік як

аргументи функцій (віддалених процедур), а в інший — як результати виконання функцій (віддалених процедур);

– технологія CORBA, що проектує варіант взаємодії між процесами, як між об'єктами CORBA, та є продовженням розвитку технології RPC.

Аналізуючи ці взаємодії більш детально, ми бачимо, що вони можуть застосовувати програмну взаємодію або як її ще називають — взаємодію між додатками.

Попередній аналіз предметної області свідчить, що для задовільнення її умов потрібно реалізувати наступних завдання:

- авторизація користувачів;
- зручне середовище, де б була можливість редагувати та створювати питання;
- збереження та оголошення тестів;
- можливість управляти та редагувати створені тести;
- можливість адміністратора передивлятися дані всіх колекцій;
- можливість адміністратора управляти в системі користувачами, які в ній зареєстровані.

Після узагальнення вимог можна підвести такий підсумок: веб-додаток буде мати зручний інтерфейс для користувачів та буде доступний із будь якого девайсу із доступом до інтернету.

2 АНАЛІЗ ПРОБЛЕМ РОЗРОБКИ WEB-ІНТЕРФЕЙСУ СТЕНДУ МОДЕЛЮВАННЯ ВІДДАЛЕНОГО МОНІТОРИНГУ ТА УПРАВЛІННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ

Можливість працювати з комп'ютером дистанційно зручно та іноді дуже потрібно.

Програми віддаленого (дистанційного) адміністрування — програми або функції ОС, завдяки яким можна мати віддалений доступ до комп'ютера через Інтернет та забезпечити управління та адміністрування віддаленого комп'ютера в реальному часі.

Дистанційне управління комп'ютером через мережу Інтернет — це робота на комп'ютері з будь-якої точки світу так, як ми б могли працювати сидячи за ним. В місцях, де є мережа Інтернет, можна здійснити підключення до комп'ютера і користуватися ним, використовуючи всі його технічні можливості для розв'язання різних завдань.

Головні задачі, що постають при вдосконаленні і впровадженні автоматизованих систем для керування технологічними об'єктами, є задоволення всіх потреб для виконання безперервного моніторингу, контролю та дистанційного керування технологічним процесом. Ця задача стає найбільш важливою тоді, коли об'єкти, з якими працює система можуть бути на великій відстані територіально один від одного, або у важкодоступних або небезпечних для людини місцях.

Основною умовою до засобів моніторингу, є мобільність, яка має бути задовільнена, тобто потрібно забезпечити доступ до потрібної інформації у будь-який час та з будь-якого місця за допомогою різних технічних засобів. Для цього моніторингу застосовують різні технології: безпроводні мережі, мережі датчиків, мобільні пристрої, супутникову навігацію, хмарні технології та сховища даних. Знаряддя моніторингу повинні гарантувати спостереження і прогнозування всіх

етапів процесу для покращення ефективності додаванням всіх необхідних поправок у реальному часі.

Головною перевагою моніторингу є задовільнення безперервного контролю за станом об'єкта та можливістю долучення технологій прогнозування та інтелектуального аналізу даних для удосконалення технологічних процесів і функціонування взагалі.

На теперішньому етапі розвитку технічних та інформаційних систем є потреба в покращенні технологій дистанційного керування технічними об'єктами з застосуванням мережі Internet.

2.1 Складові упровадження моніторингу технологічних процесів

Для застосування моніторингу технологічних процесів потрібно розглянути та зпроаналізувати дві його складові: апаратну та програмну.

До апаратної частини належать пристрої таких напрямів: засоби для накопичення та збереження інформації про сам технологічний процес — давачі, пристрої, які будуть одержувати зібрану інформацію, опрацьовувати її та переводити на інші рівні управління. Рекомендується також використовувати як дротові, так і бездротові технології для взаємодії між цими складниками. Апаратні складники, які застосовують для збору та попередньої обробки даних, мають задовільняти можливість вирішення задач у реальному часі. Окрім цього, є ще багато вимог, які повинні бути враховані під час відбору апаратних складників: малогабаритність, тому що комп'ютерні складники можуть перебувати біля давачів і виконавчих механізмів, у свою чергу великі вимоги до споживчої потужності та надійності. Зменшення малогабаритних характеристик, енергоспоживання, а також підвищення надійності та швидкодії комп'ютерних компонентів можна досягти, якщо реалізувати їх у вигляді систем на кристалі (SoC) та надвеликих інтегральних схем (НВІС). Програмна складова рішення повинна незмінно сприяти високонавантаженню обчислення, працювати за принципом асинхронності та бути структуровано з використанням відкритого програмного забезпечення.

Розробляючи моніторинг, необхідно звернути увагу на вже готові апаратні та програмні засоби, тому що розроблення та виготовлення нових потребує великих коштів і часу, а запозичення основи засобів дає можливість створити практично новий продукт без великих вкладень. Під час вибору складників необхідно враховувати безліч факторів, а саме: інформацію про вже створені апаратно-програмні компоненти, їх технічні характеристики, проаналізувати співвідношення інтерфейсів до стандартів, можливості їх купівлі тощо.

Розроблення та вдосконалення засобів моніторингу повинно ґрунтуватися на таких принципах:

- системної інтеграції, що використовує як горизонтальну, так і вертикальну інтеграцію компонентів моніторингу технологічного процесу;
- інтеграції комп'ютерних, комунікаційних і програмних компонентів;
- модульності, який прогнозує використання функціонально завершених складників з виходом на стандартний інтерфейс;
- відкритості, за яким запропоноване рішення очікує можливості нарощування та оновлення функцій;
- сумісності, який прогнозує використання стандартних провідних і безпроводних інтерфейсів для зв'язку між компонентами.

Під час моніторингу роботи, зокрема технологічних процесів, утворюється первинна інформація, яка спочатку опрацьовується, накопичується та надходить на засоби контролю. Ці відомості можуть бути використані для формування сигналів для прийняття різноманітних управлінських рішень. Зробивши аналіз можна навести ряд особливостей, які потрібно враховувати під час розроблення рішення моніторингу технологічних процесів:

- бажано застосовувати технології прості, розгорнуті та з підтримкою;
- архітектура рішення має бути відкрита, з можливістю масштабування, оскільки кількість точок вимірювання параметрів може змінюватися;
- апаратна частина системи повинна розроблятися на основі сучасних типових рішень для забезпечення простоти та низької вартості;

– необхідне широке застосування базових телекомунікаційних протоколів.

За принципом роботи технічною основою для моніторингу можуть бути безпроводні сенсорні мережі. Оскільки мережі такого типу не є новинкою, вони все більше використовуються для прийняття рішень у різних сферах людської діяльності, де задіяні сенсорні мережі. В контексті інтернету сенсорні мережі володіють такими важливими властивостями, як самоорганізація і адаптивність до змін навколишніх умов та інфраструктури, а апаратне забезпечення бездротових вузлів і протоколи мережевої взаємодії між ними оптимізовані за енергоспоживанням для забезпечення тривалого терміну експлуатації системи з автономними джерелами живлення.

В інтелектуальних сенсорних системах функції сенсорів і їх інтерфейсів об'єднано в одну систему, яка здійснює попередню трансформацію параметра середовища, утворення електронного сигналу, аналого-цифрове перетворення, сполучення з шиною і опрацювання даних. У цей комплекс також можуть входити функції вищого ієрархічного рівня, наприклад, автокалібрування, самотестування, оцінка та ідентифікація даних. Для сенсорної системи, в якій задіяні функціонально складні компоненти, вживання пам'яті в мікроконтролері дає можливість накопичувати дані, які надійшли від декількох сенсорів, певного періоду. Це можливо в мікроконтролері, персональних комп'ютерах та сенсорних інтерфейсах

Структура, яка описана вище, а також принцип роботи безпроводних сенсорних мереж певною мірою зараджують вирішенню певних проблеми моніторингу технологічних процесів. Але щоб дотриматися всіх вимог, які поставлені перед розробкою, потрібно ввести деякі зміни в структуру та принцип роботи цих мереж.

Аналітична підсистема утворюється зі сховища технологічних даних, засобів ETL, аналітичного та інтелектуального упорядкування даних, прогнозування та моделювання і підтримки прийняття рішень.

Фреймворк (software framework) суттєво полегшує створення програмного забезпечення. Зараз засоби передавання даних формують на основі технологій Ethernet та Wi-Fi, що викликано тим, що кабельних системи є доступними та мають

широку номенклатуру, активного і пасивного мережевого устаткування. Технологія безпроводних мереж Wi-Fi найвигідніша, якщо є вимоги щодо мобільності, елементарності встановлення і застосування. Великим плюсом є гнучкість архітектури мережі з перспективою до динамічної зміни топології мережі, швидкість проектування і реалізації.

Для того щоб створити мікропрограму потрібно задіяти доступний програмний фреймворк, з використанням бібліотеки підтримки широкого кола давачів. Для розробки програми можна використати мову одну із найпопулярніших на сьогодні мов — Java.

Зараз каталог одноплатних комп'ютерів на SoC дуже розширився. Всі вони мають розвинену периферію, також до них можна приєднати периферійні технічні засоби — різні види датчиків, а також забезпечити роботу в безпроводній мережі Internet.

Але при опрацюванні матеріалу за наявності великого різноманіття різноманітних апаратних засобів виникає питання, як об'єднати всі присутні апаратні засоби в одну інфраструктуру та зробити зручні інтерфейси для них. Одним із варіантів є розроблення та імплементація власного рішення, але, якщо взяти до уваги всі вимоги до нього, можна підсумувати, що для цього потрібно дуже багато фінансових та часових ресурсів. Тому, проаналізувавши весь спектр проблеми, ми знайшли багато готових рішень для вирішення цієї проблеми.

2.2 Віддалений моніторинг та керування технологічними об'єктами за допомогою web-технологій

Система віддаленого моніторингу представляє собою спеціалізований додаток, який наглядає за комп'ютером та його керуванням, коли користувач перебуває на певній відстані від нього.

Роль цієї система полягає в тому, щоб визначити працездатність персонального комп'ютера та коректування проблем, які можуть з'явитися, без потреби фізичного доступу до машини.

Основними рисами систем моніторингу, які популярні, є можливість проглядати інформацію про комп'ютер у зручній адаптованій формі. Це дає можливість використовувати ці програми не тільки професіоналам, а й будь-якій людині. Ця система зможе виконувати безліч команд на стороні серверу, а це дозволить спростити процес управління віддаленим комп'ютером.

Здійснення керування та моніторингу всіляким технічним пристроєм, застосовуючи браузер та локальну мережу або мережу Internet, повинно відповідати наступним основним вимогам:

- застосування системи повинно незначно навантажувати інформаційну мережу;
- обмін даними між інтерфейсом та адаптером повинен здійснюватись із підтримкою стандартного web-протоколу HTTP, це усуне потребу в налаштуванні мережевого обладнання;
- всі данні, які реконструюються інтерфейсом, мають активно змінюватись у вікні браузера без потреби повторного перезавантаження файлів інтерфейсу;
- вартість фізичного адаптера повинна варіювати межах відповідних аналогів.

Реалізація функцій, які були вказані, потребує апаратних і програмних рішень (рисунок 2.1).

Апаратна частина репрезентована у вигляді спеціальної схеми - «адаптера» на базі мікроконтролера, який трансформує web-запити в управляючі команди. «Адаптер» також має утворювати текст відповіді, яка підкріплюється виконанням операції та може мати додаткову інформацію про стан пристрою. Програмна частина має вигляд вихідних кодів інтерфейсу, бібліотек функцій та програмного забезпечення мікроконтролера адаптера.



Рисунок 2.1 Загальна структура віддаленого доступу з використанням мережі Інтернет

«Web-сервером» може являтися будь-який девайс з спеціальним програмним забезпеченням, яке опрацьовує web-запити та створює відповіді на них. А приклад може слугувати така програма як «Apache». Вона працює на Windows та Unix-подібних операційних системах. Комп'ютер зберігає файли інтерфейсу, які потім завантажуються браузером.

Любий пристрій, який під'єднаний до цієї системи є «Комп'ютером», з вимогою того, що він з'єднаний з мережею Internet та має web-браузер (Internet Explorer, Opera, Safari, Chrome і т.п.). Отже система, яка структурована таким чином, може діяти як у локальній мережі, так і в мережі Інтернет.

Щоб розпочати працювати із системою потрібно через «Браузер» завантажити «web-інтерфейс», після того як сторінка інтерфейсу відкривається запускається web-додаток. Наступний крок, це встановлення зв'язку з адаптерною платою, вона з'єднана з локальною системою керування фізичним об'єктом. Потім, коли встановлюється зв'язок, запускається функціональне ядро додатку, де періодично створюються web-запити до адаптера й перевантажується нова інформація про стан об'єкта керування. За запитом користувача або чере якийсь період часу вся інформація оновлюється.

Коли потрібно робити якісь дії, внести зміни або передати дані, web-додаток створює запит та передає його на адаптер, який потім формує звіт про результати виконання запиту й перенаправляє його назад до додатку. Результати, які знову

отримує додаток аналізуються й висвічуються на екрані. Робота з такою структурою програми та використання цього алгоритму роботи дають можливість завантажувати інтерфейс тільки один раз, завдяки цьому значно зменшується трафік, а це не так навантажує мережу.

Програмне забезпечення мікроконтролера апаратного адаптера має такий склад: інтерфейси обміну даними, список властивостей, функціональне ядро та конфігурування ядра, низькорівневий драйвер обміну даними та USB-драйвер.

Інтерфейс обміну даними приймає команди та відправляє всі дані через спеціальний драйвер.

Список властивостей містить в собі список об'єктів, які формуються з імені властивості та її фізичної адреси. Цей принцип покращує функціонування всієї системи, тому що web-інтерфейс звертається саме до іменованої властивості, а не до фізичної адреси порту мікроконтролера, тобто це підвищує якість роботи всієї програми.

Функціональне ядро являється основним елементом програми, яка під час запуску мікроконтролера починає виникати та відповідає за утворення всіх інших об'єктів, які потрібні для функціонування системи та складається з основних алгоритмів роботи програми.

Інтерфейс конфігурування ядра — базовий інтерфейс, він складається з цілого набору функцій, які потрібні для конфігурування властивостей ядра та роботи головних функцій.

Низькорівневий драйвер обміну даними — створює зв'язок з програмованим контролером локальної системи керування фізичного об'єкту. Цей драйвер можна прописати для всіх інтерфейсів обміну даними. Також є можливість того, щоб він був віртуальним і не мав контактів з фізичним світом (для налагоджувальних цілей). Патерн «драйвер - інтерфейс» дає можливість програмі бути гнучкою та незалежною від платформи, на якій вона реалізується.

USB-драйвер — підтримує зв'язок між контролером та комп'ютером, через шину USB.

Щоб програма була більш гнучкою використовується об'єктно-орієнтоване програмування. Якщо архітектура побудована правильно, то можна вільно переносити програму на різні апаратні платформи (AVR, ARM, PIC і т.ін.) лише вносячи зміни в низько-рівневі функції та об'єкти. Для того, щоб перевірити працездатність системи, за допомогою емулятора електронних схем була створена тестова схема, яка працює використовуючи мережу Internet. Головне вікно додатку та віртуальна електронна схема зображені на рисунки 2.2. та 2.3.



Рисунок 2.2 Інтерфейс користувача

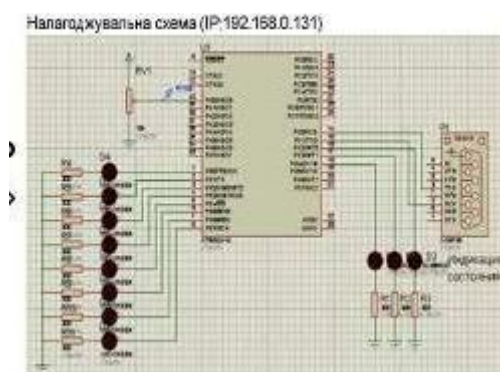


Рисунок 2.2 Схема керування

Отже, роблячи аналіз системи віддаленого моніторингу, можна зробити такий висновок: система, яка навми виготовлена, вживає значно менше системних ресурсів, ніж додатки, які були заявлені раніше, тому що вона не потребує доступу до робочого столу комп'ютера.

Основними вимогами, які ставляться користувачами до систем моніторингу є:

- зрозумілий інтерфейс та вигідна система навігації;
- надійність;
- можливість кастомізувати інтерфейс.

З розвитком інформаційних технологій системи віддаленого моніторингу зайняли чільне місце в співпраці людини комп'ютером. Ця тема є дуже актуальною на даний момент, тому що віддалений моніторинг та управління дають можливість користувачу слідкувати за сервером без фізичного доступу.

2.3 Існуюча реалізація віддаленого моніторингу та управління технологічним процесом

Розумна web-система підтримує безліч проектів. Але самий перспективний, на даний час, це система розумний будинок.

Система розумний будинок — це автоматизована система управління, яка контролює і управляє усіма інженерними системами будинку, таких як: електропостачання, опалення, кондиціонування, системи безпеки, мультимедіа та ін.

Всі пристрої, які є в системи розумний будинок, такі як комп'ютери, планшети, мобільні телефони та інші, створюють інформаційну мережу, де проходить обмін даними між вузлами системи. Управління системою розумний дім за допомогою Інтернету має віддалений контроль.

Нещодавно систему розумний дім мали одиниці, вона вважалася ознакою багатства власника житла, адже мала високу вартість. Багато коштувало як саме обладнання, так і його програмне забезпечення. Але після початку розвитку ІТ технологій: технологій автоматизації, каналів зв'язку, а також мобільних пристроїв все змінилося.

З кожним роком системи розумний будинок стають все більш доступними, але до них встановлюються нові критерії, все більше вимогливості до комфортної життєдіяльності. Власники нерухомості тепер хочуть, щоб система була не лише функціональна і зручна, а й їх ергономічна, практична і надійна. Тому зараз

системи розумний дім проектується так, щоб задовольняти саме ці потреби користувачів.

В результаті аналізу технологій, на базі яких реалізована система управління розумний будинок можна виділити:

1. Управління через Інтернет

Для управління і налаштування системи Розумний Дім віддалено тепер можна за допомогою електронної пошти, а саме передавати через неї всі необхідні команди. Для виконання цієї умови основна програма поділяється на два незалежних модуля, один з них перебуває в будинку та налаштований для отримання команд. Інший модуль знаходиться на пристрої користувача.

2. Дистанційне керування

Щоб управління побутовими пристроями в будинку зробити зручним, автори розробили пульт дистанційного керування, який може управляти телевізором, відеомагнітофоном, музичним центром, супутниковим ресивером. Цей пульт також дає можливість вмикати або вимикати: освітлювальні прилади, керовані електричні розетки та інше. Для цього на ньому запрограмовано комбінації декількох кнопок, що дає змогу виконувати багнато дій, таких як, відкрити ворота, поставити будинок на охорону.

3. Управління з комп'ютера

Програма, яка працює в середовищі операційної системи Windows, дає можливість включати і виключати певні режими системи Розумний Дім, робити налаштування її роботи, читати і виводити на друк протокол повідомлень.

Головним в цій системі є центральний блок управління. Комп'ютер робить систему універсальною, гнучкою, простотою у використанні. За допомогою комп'ютера можна задавати величезну кількість різних завдань в рамках однієї системи.

4. Управління з планшета

Переважно всі системи розумного будинку роблять в автоматичному режимі і не потребують втручання людини. Однак іноді виникає потреба повідомити користувачеві якусь інформацію, яка йому була б корисною: температура на вулиці,

прогноз погоди, зображення з камер спостереження та інше. Крім того, іноді користувачу потрібно дистанційно керувати деякими елементами, наприклад, світлом, побутовими приладами, вносити зміни в роботу кліматичних і охоронних модулів.

Тобто, Розумний Будинок складається ніби з трьох основних елементів: центральний процесор, керовані їм виконавчі механізми та інтерфейсні пристрої.

Центральний процесор – це сервер. Для нього можна використовувати все, що завгодно, від суперкомп'ютера з гелієвим охолодженням до роутера і мікроконтролера.

5. Управління з промислового логічного контролера (ПЛК)

Як стандартна автоматизована система, система Розумний будинок створена за трирівневим принципом: нижній рівень (датчики температури, силові контактори і реле), на середньому рівні програмований логічний контролер, модулі введення-виведення та GSM-модем. Верхній рівень (HMI, SCADA) складається з панель оператора і серверного комп'ютера, на якому є web-інтерфейс.

При розробці системи управління розумний дім потрібно проаналізувати системи життєзабезпечення будівлі як об'єкта управління.

Проаналізувавши існуючих технологій, які використовують для створення систем типу розумний дім, ми прийшли до висновку, що перспективним є створення систем управління життєзабезпеченням будівель не тільки на базі контролерів, але з доповненням їх персональним комп'ютером або планшетом. Завдяки цьому розумний будинок може управляти не тільки комунікацією, кліматом і обладнанням, а й середовищем для обміну і трансформацією даних, медіа-сервером, контент-сервером і при наявності web-інтерфейсу система на базі комп'ютера є перспективним і цікавим рішенням.

Висновки до розділу 2

Розглянуто існуючі впровадження віддаленого моніторингу та управління технологічним процесом. При аналізі з'ясовано, що головною перевагою

моніторингу є не лише гарантія постійного контролю за станом об'єкта, але й можливість залучати технології інтелектуального аналізу даних, щоб покращити технологічних процесів і функціонування системи.

3 WEB-ІНТЕРФЕЙС

Web-інтерфейс має адміністративну частину, в ній проводяться налаштування клієнтської частини, надаються права для перегляду інформації по об'єктах користувачам, а також дається можливість змінювати дані лічильників та об'єктів диспетчеризації, переглядати та виправляти архіви даних лічильників, управляти роботою запису та зчитування лічильників. Є два типи доступу до web-інтерфейсу: звичайний — користувач повинен авторизуватися і тоді йому надається можливість переглядати дані про об'єкти, але він не має доступу до адміністрування; повний — користувач має всі права доступу до web-інтерфейсу, як до користувацької частини, так і до адміністративної, що дає можливість налаштовувати web-інтерфейс.

3.1 Web-інтерфейс

Інтерфейс — це засіб візуального інформаційного спілкування між користувачем і програмою. Web-інтерфейс — засіб комунікації, який проводить взаємодію з користувачем через комп'ютерну мережу. Дуже часто можна почути, що web-інтерфейс розуміють як віддалений інтерфейс, який ґрунтується на гіпертекстовому протоколі HTTP, адже він самий поширений у всесвітній павутині протоколів передачі даних.

Коли користувач використовує web-інтерфейс він взаємодіє з програмою а тим же принципом, як зі звичайним web-сайтом. Адже звичайний web-сайт також є web-інтерфейсом, який має специфічне завдання — надавати користувачеві необхідну інформацію. Коли ж користувач використовує спеціалізований web-інтерфейс, він може виконувати практично всі завдання, які виконує взаємодіючи зі звичайним інтерфейсом, як приклад можна назвати управління яким-небудь пристроєм.

Порівнюючи web-інтерфейс зі звичайним програмним можна обізначити наступні переваги:

- програмне забезпечення не потрібно завантажувати та встановлювати включаючи компоненти, необхідні для його функціонування;

- користувач не зможе скачати і встановити вірусне програмне забезпечення;

- спрощується завдання адміністрування/обслуговування великої кількості робочих станцій. Скорочується кількість локальних несправностей програм, які пов'язаних зі зміною системи, пошкодженням файлів;

- все програмне забезпечення оновлюється на сервері самостійно, отже користувачу не потрібно стежити а новинками;

- простіша організація колективної роботи;

- можливість шифрування переданої інформації (протокол HTTPS, ключі SSL);

- кроссплатформенність, незалежність від характеристик системи користувача.

Але web-інтерфейси мають також і недоліки:

- збільшуються витрати мережевого трафіку;

- завантаження інтерфейсу в клієнт/браузер користувача, що є певною проблемою;

- витрати на обслуговування серверної частини збільшуються, виходячи з попереднього пункту;

- проектування інтерфейсу потрібно вести з урахуванням різних можливостей, а також неможливість коректного відображення браузерами.

Недоліки, які приведені нижче відносяться взагалі до всіх мережних додатків на базі клієнт-сервера, але іноді (залежачи від завдань програми) вони відносяться до web-інтерфейсів в тому числі:

- недоступність інтерфейсу при проблемі з мережею, зв'язком;

- не досягається необхідна швидкість передачі даних по мережі, якщо система потребує великої швидкості відгуку користувача, а це знижує ефективність системи.

- можливість втрати інформації, пов'язаної з мережевим збоєм.

Як ми бачимо переваги в застосуванні web-інтерфейсів для комп'ютерних програм значно переважають їх недоліки, це говорить про те, що їх застосування в подальшому майбутньому розвитку інформаційних технологій є дуже

перспективним. Цей висновок також можна зробити дослідивши досить успішними приклади їх практичного застосування. А рівень розвитку технологій, на яких самі інтерфейси базуються, дозволяє використовувати програми, які можуть виконати практично всі завдання, від налаштування мережевого обладнання до управління складними автоматизованими системами.

На даний момент web-інтерфейси вже активно застосовуються в самих різних областях діяльності, пов'язаних з ІТ.

3.2 Інтерфейс користувача

Інтерфейс користувача (UI - англ. *user interface*) — різновид інтерфейсів, в якому один бік закріплений за людиною (користувачем), інша - машиною/пристроєм. Інтерфейс користувача — це сукупність засобів і методів, які надають можливість користувачу взаємодіяти з різними машинами, пристроями і апаратурою.

Доволі часто цей термін застосовується по відношенню до комп'ютерним програмам, але він може мати на увазі набір засобів, методів і правил взаємодії будь-якої системи, керованої людиною.

Кілька широко поширених прикладів:

- меню на екрані телевізора + пульт дистанційного управління;
- дисплей електронного апарату + набір кнопок і перемикачів для настройки;
- приладова панель + важелі управління.

Інтерфейс вважається інтерактивним, коли пристрій, який отримав команду від користувача, виконує її та надає інформацію користувачеві різними засобами – візуальними, звуковими, тактильними тощо (приймавши інформацію користувач надає інші команди такими засобами, як: кнопки, перемикачі, регулятори, сенсори, голосом, і т. д.

Оскільки інтерфейс сам по собі є складним елементом, то він складається з елементів, які також можуть складатися з елементів (так, екран може містити в

собі декілька вікон, які також можуть мати панелі, кнопки та інші інтерфейсні елементи.

Але головним в інтерфейсі користувача традиційно являється його ефективність та зручності використання. Він повинен бути зрозумілий та зручний.

Під сукупністю засобів і методів інтерфейсу користувача маються на увазі:

Засоби:

- виведення інформації з девайсу до користувача — через будь-які впливи на організм людини (зорових, слухових, тактильних, нюхових і тд.) - екрани (дисплеї, проектори) і лампочки, динаміки, зумери і сирени, вібромотор і тд. і тп.

- введення певних команд користувачем в пристрій через кнопки, перемикачі, потенціометри, датчики положення та руху, сервоприводи, жести особою і руками, навіть через мозкову активності користувача.

Через ввиди засобів введення, інтерфейси поділяються на типи — жестові, голосові та інші, є можливість застосування змішаних варіантів. Всі варіанти повинні бути необхідними, зручними і практичними, розташованими і скомпонованими зрозуміло та оптимальним набором елементів, відповідати фізіології людини та не вносити негативні наслідки для організму користувача.

Методи:

- всі правила та методи, які закладені розробником пристрою, повинні привести до необхідної реакції пристрою при запиті користувачем і виконанню необхідної задачі.

Всі правила та методи повинні бути зрозумілими та легкими для запам'ятовування (все це входить в поняття юзабіліті).

Збільшення у пристрої методів вводу-виводу дає можливість спростити методи управління та правил користування, але є і інший бік, це призводить до складності сприйняття інформації користувачем — інтерфейс стає перевантаженим. Навіть, якщо зменшити навантаження на засоби відображення і контролю все рівно, це приводить до ускладнення правил управління

- кожен елемент бере на себе дуже багато функцій. Тому проєктувальники інтерфейсів пробують знайти компромісне рішення між цими двома крайностями в кожному окремому випадку.

Інтерфейс користувача комп'ютерного додатка включає:

- засоби відображення інформації;
- командні режими, мова "користувач - інтерфейс";
- пристрої та технології введення даних;
- взаємодія між користувачем і комп'ютером та зворотний зв'язок з користувачем;
- підтримку прийняття рішень у конкретній предметній області;
- порядок використання програми та документацію на неї.

Часто вважають, що інтерфейс користувача, це зовнішній вигляд програми. Але це неправильно. Користувач працює з ним як з усією програмою в цілому, тобто попереднє ствердження є занадто вузьким.

Насправді інтерфейс користувача об'єднує в собі всі елементи і компоненти програми, які впливають на роботу користувача з програмним забезпеченням, це не лише екран ПК.

До цих елементів відносяться:

- набір завдань користувача;
- використовувана системою метафора (наприклад, робочий стіл в MS Windows);
- елементи керування системою;
- навігація;
- дизайн екранів програми;
- засоби відображення інформації, яка відображається інформація та формати;
- пристрої та технології введення даних;
- діалоги, взаємодія і транзакції між користувачем і комп'ютером;
- зворотній зв'язок з користувачем;
- підтримка прийняття рішень у конкретній предметній області;

- порядок використання програми і документація на неї.

-

Висновки до розділу 3

Web-інтерфейс також має адміністративну частину, в якій можна проводити налаштування клієнтської частини, надання різних прав для перегляду інформації по об'єктах користувачам, змінювати дані лічильників та об'єктів диспетчеризації, редагувати архіви даних лічильників, керувати роботою запису та зчитування лічильників. Існує два типи доступу до web-інтерфейсу:

- звичайний — після авторизації користувач може переглядати дані про об'єкти, але не має доступу до адміністративної частини;
- повний — користувач має максимальні права доступу до web-інтерфейсу як до користувацької частини, так і до адміністративної для налаштування web-інтерфейсу.

4 ОБГРУНТУВАННЯ ВИБРАНИХ ЗАСОБІВ РОЗРОБКИ

Для якісного і об'єктивного вибору мови програмування, потрібно порівняти найбільш актуальні мови програмування та фреймворки. Після цього визначити всі переваги і недоліки кожної з мов програмування. На сучасному розвитку найбільш актуальним і прогресивними мовами програмування є мова Java та фреймворк JetBrains IntelliJ IDEA.

4.1 Обґрунтування вибору мови програмування та фреймворку

Java 8EE - скриптова мова, призначена для створення інтерактивних веб-сторінок, проте на цьому сфера її застосування не закінчується. Її активно використовують при розробці серверної частини веб-додатку чи навіть ігор.

Програми на Java транслуються в байт-код, що виконується віртуальною машиною Java (JVM) - програмою, обробній байтовий код і передавальній інструкції обладнанню як Лари.

Гідність подібного способу виконання програм — в повній незалежності байт-коду від операційної системи і обладнання, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання.

Часто до недоліків концепції віртуальної машини відносять те, що виконання байт-коду віртуальною машиною може знижувати продуктивність програм і алгоритмів, реалізованих на мові Java. Останнім часом було внесено ряд удосконалень, які кілька збільшили швидкість виконання програм на Java:

- застосування технології трансляції байт-коду в машинний код безпосередньо під час роботи програми (JIT -технологія) з можливістю збереження версій класу в машинному коді,
- широке використання платформенно-орієнтованого коду (native-код) в стандартних бібліотеках,
- апаратні засоби, що забезпечують прискорену обробку байт-коду (наприклад, технологія Jazelle, підтримувана деякими процесорами фірми ARM).

Основні можливості JAVA:

- автоматичне керування пам'яттю;
- розширені можливості обробки виняткових ситуацій;
- багатий набір засобів фільтрації вводу / виводу;
- набір стандартних колекцій, таких як масив, список, стек і т. п.;
- наявність простих засобів створення мережових додатків (у тому числі з використанням протоколу RMI);
- наявність класів, що дозволяють виконувати HTTP -запити і обробляти відповіді;
- вбудовані в мову засоби створення багатопотокових додатків;
- уніфікований доступ до баз даних :
- на рівні окремих SQL -запитів — на основі JDBC, SQLJ;
- на рівні концепції об'єктів, що мають здатність до зберігання в базі даних - на основі Java Data Objects (*англ.*) і Java Persistence API (*англ.*);
- підтримка шаблонів (починаючи з версії 1.5);
- паралельне виконання програм.

За допомогою Java 8EE можна виконувати наступні дії:

- змінювати сторінку, писати на ній текст, додавати і видаляти теги, міняти стилі елементів;
- реагувати на події: скрипт може чекати, коли щось станеться (клік, закінчення завантаження сторінки) і реагувати на це виконанням функції;
- виконувати запити до сервера і завантажувати дані без перезавантаження сторінки. Це іноді називають "AJAX";

- встановлювати і зчитувати cookie, валідувати дані, виводити повідомлення і багато іншого.

До сильної сторони даної мови належить повна підтримка браузерами. Наприклад, такі технології як ActiveX, VBScript, XUL — підтримуються не в кожному браузері. Такі технології як Flash, Silverlight, Java — не повністю інтегровані з браузером, працюють в своєму оточенні.

Дана мова програмування не позбавлена недоліків, таких, як, наприклад, нестрога типізація, що є причиною неявного переведення типів. Однак усі недоліки перебиваються поширеністю та кросбраузерністю мови.

Для більш продуктивного і простого процесу розробки було вибрано фреймворк Vaadin, в якому використовується мова Java 8EE як на клієнті, так і на сервері, а також в роботі з базою даних, що дозволяє виконувати один і той самий код на фронт-енді та бек-енді одночасно.

Vaadin — вільно поширюваний фреймворк для створення RIA-веб-додатків, що розробляється однойменної фінської компанією. На відміну від бібліотек на Javascript і специфічних плагінів для браузерів, Vaadin пропонує сервер-орієнтовану архітектуру, яка базується на Java Enterprise Edition. Використання JEE дозволяє виконувати основну частину логіки програми на стороні сервера, тоді як технологія AJAX, яка використовується на стороні браузера, дозволяє інтерактивно взаємодіяти з користувачем, не відстаючи від аналогічних десктоп-додатків. Для відображення елементів призначеного для користувача інтерфейсу і взаємодії з сервером на стороні клієнта Vaadin використовує Google Web Toolkit.

Використання Java як єдиної мови програмування при створенні веб-додатків і веб-контенту — одна з найбільш значущих функцій в Vaadin. Фреймворк використовує подієву модель і певні елементи призначеного для користувача інтерфейсу, віджети, що робить її дуже близькою до моделі розробки настільних додатків на Java з використанням HTML і Javascript.

Організація моделі даних і віджетів дозволяє відображати в браузері великі обсяги даних без значної завантаження оперативної пам'яті і без додаткових дій з

боку розробника. Використання Google Web Toolkit для відображення сторінок з результатами пошуку і обробки дій користувача (на зразок термінального клієнта). Так як Google Web Toolkit функціонує тільки на стороні клієнта, Vaadin додає додаткову затвердження даних на стороні сервера: це вирішує проблеми безпеки, пов'язані з можливістю підміни даних або коду Javascript. Відповідно, при зміні і пошкодженні даних, що надходять від браузера, сервер, визначивши це, не пропускає запити.

Можливість розширення забезпечується можливістю використання додаткових віджетів, написаних для GWT, а також кастомізації за допомогою CSS. Однак стандартний додаток, що створюється на Vaadin, не вимагає програмування саме на GWT та подальшої компіляції GWT-компілятором, якщо тільки розробник не додає в проект нестандартні віджети.

Hypertext Markup Language (HTML) (укр. *Мова розмітки гіпертекстових документів*) — стандартна мова розмітки для створення веб-сторінок і веб-додатків. З Cascading Style Sheets (CSS) і JavaScript, вона утворює тріаду основних технологій для World Wide Web.

Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML окреслені *тегами*, написаними з використанням кутових дужок. Теги, такі як і безпосередньо вводять вміст на сторінку. Інші теги, такі як `` `<input />` `<p>` оточують і надають інформацію про текст документа і можуть включати інші теги як під-елементи. Браузери не показують теги HTML, але використовують їх для інтерпретації вмісту сторінки.

HTML може вбудовувати програми, написані на мові сценаріїв, наприклад JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і компоновання вмісту. World Wide Web Consortium (W3C), які супроводжують як HTML і CSS стандартів, заохочує використання CSS над явним презентаційним HTML з 1997 року.

HTML впроваджує засоби для:

- створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

4.2 Обґрунтування вибору середовища розробки

При виборі середовища розробки для веб-додатку імітаційного моделювання обчислювальних систем було розглянуто наступні середовища:

JetBrains WebStorm — інтегроване середовище розробки для JavaScript, HTML та CSS від компанії JetBrains, розроблена на основі платформи IntelliJ IDEA.

Інтегроване середовище розробки (ICP, англ. Integrated development environment або англ. IDE) — комплексне програмне рішення для розробки програмного забезпечення. Зазвичай, складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм.

WebStorm є спеціалізованою версією PhpStorm, пропонуючи підмножину з його можливостей. WebStorm постачається з перед-установленим плагінами JavaScript (такими як для Node.js), котрі доступні для PhpStorm безоплатно.

WebStorm забезпечує автодоповнення, аналіз коду на льоту, навігацію по коду, рефакторинг, зневадження та інтеграцію з системами управління версіями.

Налáгодження програ́ми, в мережі рідше знева́дження (англ. debugging) — методичний процес пошуку та зменшення числа помилок або дефектів у

комп'ютерній програмі або електронному обладнанні з метою отримання очікуваної поведінки.

Важливою перевагою інтегрованого середовища розробки WebStorm є робота з проектами (у тому числі, рефакторинг коду JavaScript, що міститься в різних файлах і теках проекту, а також вкладеного в HTML). Підтримується множинна вкладеність (коли в документ на HTML вкладений скрипт на Javascript, в який вкладено інший код HTML, всередині якого вкладений Javascript) — в таких конструкціях підтримується коректний рефакторинг.

Основні можливості:

- Модифікація файлів .css, html, .js з одночасним переглядом результатів (Live Edit, в деяких джерелах ця функціональність називається «редагування файлів на льоту» або «в реальному часі» або «без перезавантаження сторінки»)
- Підтримка HTML5
- Підтримка JSDoc
- Підтримка Node.

LiveEdit — можливість WebStorm, котра з'явилася з версії 5 і дозволяє одночасно редагувати код html, css або javascript і бачити, як результат відображається в браузері. Для цього потрібна підтримка такої можливості з боку браузера, тому WebStorm при установці ставить плагін для Google Chrome.

Subversion (з 2010 називається Apache Subversion) — вільна система управління версіями.

Google Chrome — веб-переглядач, розроблений компанією Google на основі веб-переглядача з відкритим кодом Chromium та іншого відкритого програмного забезпечення. Його програмна архітектура була створена з чистого аркуша (однак використовуючи інше програмне забезпечення з відкритим кодом, у тому числі компоненти WebKit та Mozilla) з метою задоволення поточних потреб користувачів.

WebStorm підтримує зневадження застосунків у node.js. Також підтримується повний набір функцій редагування застосунків на javascript — як для виконання на сервері, так і в браузері: автодоповнення, навігація по коду, рефакторинг і перевірка на помилки.

Для `node.js` підтримується також виведення повідомлень `node.js` на окрему вкладку в IDE.

Мови стилів LESS, Sass і SCSS, які розширюють можливості описів стилів у CSS, повністю підтримуються в WebStorm, зокрема, підтримується рефакторинг коду для них, коли треба змінити вираз (наприклад, `#a9a9a9`) на змінну (наприклад `@grey`), для того, щоб зробити код читанішим і простіше перевизначати параметри (наприклад, шляхом присвоєння їм значення `@grey: #a9a9a9`)

У версіях від WebStorm 5 для CoffeeScript є навігація за кодом, автодоповнення, рефакторинг, підсвічування синтаксису і перевірка на помилки.

Eclipse – вільне модульне інтегроване середовище розробки програмного забезпечення. Розробляється і підтримується Eclipse Foundation і включає проекти, такі як платформа Eclipse, набір інструментів для розробників на мові Java, засоби для управління сирцевими кодами, візуальні побудовники GUI тощо.

CoffeeScript (`[ˈkɔːfi skɪpt]`; кофі скрипт) — мова програмування, що транслюється в JavaScript. CoffeeScript додає синтаксичний цукор у дусі Ruby, Python і Haskell для того, щоб покращити читання коду і зменшити його розмір.

Eclipse Foundation — некомерційна організація, що координує розробку Eclipse.

Написаний в основному на Java, може бути використаний для розробки застосунків на Java і, за допомогою різних плагінів, на інших мовах програмування, включаючи Ada, C, C++, COBOL, Fortran, Perl, PHP, Python, R, Ruby (включно з каркасом Ruby on Rails), Scala, Clojure та Scheme. Середовища розробки зокрема включають Eclipse ADT (Ada Development Toolkit) для Ada, Eclipse CDT для C/C++, Eclipse JDT для Java, Eclipse PDT для PHP.

Початок коду йде від IBM VisualAge, він був розрахований на розробників Java, складаючи Java Development Tools (JDT). Але користувачі могли розширяти можливості, встановлюючи написані для програмного каркасу Eclipse плагіни, такі як інструменти розробки під інші мови програмування, і могли писати і вносити свої власні плагіни і модулі.

Випущена на умовах Eclipse Public License, Eclipse є вільним програмним забезпеченням. Він став одним з перших IDE під GNU Classpath і без проблем працює під IcedTea.

Eclipse це фреймворк для розробки модульних кросплатформових застосунків із низкою особливостей:

- можливість розробки ПЗ на багатьох мовах програмування (рідною є Java);
- крос-платформова;
- модульна, призначена для подальшого розширення незалежним розробниками;
- з відкритим сирцевим кодом;
- розробляється і підтримується фондом Eclipse, куди входять такі постачальники ПЗ, як IBM, Oracle, Borland.

Eclipse насамперед повноцінна Java IDE, націлена на групову розробку, має засоби роботи з системами контролю версій (підтримка CVS входить у поставку Eclipse, активно розвиваються кілька варіантів SVN модулів, існує підтримка VSS та інших). З огляду на безкоштовність, у багатьох організаціях Eclipse — корпоративний стандарт для розробки ПЗ на Java.

Eclipse написана на Java, тому є платформи-незалежним продуктом, крім бібліотеки графічного інтерфейсу SWT, яка розробляється окремо для більшості поширених платформ. Бібліотека SWT використовує графічні засоби платформи (ОС), що забезпечує швидкість і звичний зовнішній вигляд інтерфейсу користувача.

Робоче середовище Eclipse (панелі, редактори, проекції, майстри).

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .

Windows Mobile (для версій 6.x також відома як Windows Phone) - операційна система для мобільних пристроїв з основним набором програм, таких як Windows Marketplace for Mobile, My Phone, Windows Live, заснованих на Microsoft Win32 API.

NET Framework, Xbox, Windows Phone .

Windows Phone (кодова назва «Photon») — велике оновлення Windows Mobile, що вийшло 11 жовтня 2010 року.

NET Compact Framework і Silverlight.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторинга коду.

Розглянувши усі наведені середовища розробки було прийнято рішення використовувати JetBrains IntelliJ IDEA. В цьому середовищі інтегровані Git, Gulp, Mocha які використовуються в розробці веб-додатку для імітаційного моделювання обчислювальних систем. Тому, це середовище розробки є найбільш зручним.

4.3 Структура web-додатку

Додатки зазвичай діляться на логічні частини, звані "шарами", при цьому кожному шару призначається своя роль. Локальні додатки можуть складатися тільки з одного шару, який розміщується на комп'ютері клієнта, а web-додатки за своєю природою слідуєть N -слойному підходу. Хоча можливі різні варіанти, найбільш поширеними є функції, які залежать три шари: *шар уявлення* ; *шар бізнес-логіки* ; *шар доступу до даних* (сховище). Кожен шар включає набір компонент (наборів класів), які виконують спеціальні функції.

На рисунку 4.1 показана типова архітектура web-додатки з набором найбільш часто вживаних компонент, згрупованих за функціональним областям.

Шар уявлення зазвичай включає компоненти для користувача інтерфейсу (UI) і логіку уявлення. Шар бізнес-логіки включає компоненти бізнес-логіки, бізнес-процесу і бізнес-сутностей. Шар доступу до даних включає компоненти, що реалізують доступ до необхідних даних і web-сервісів.

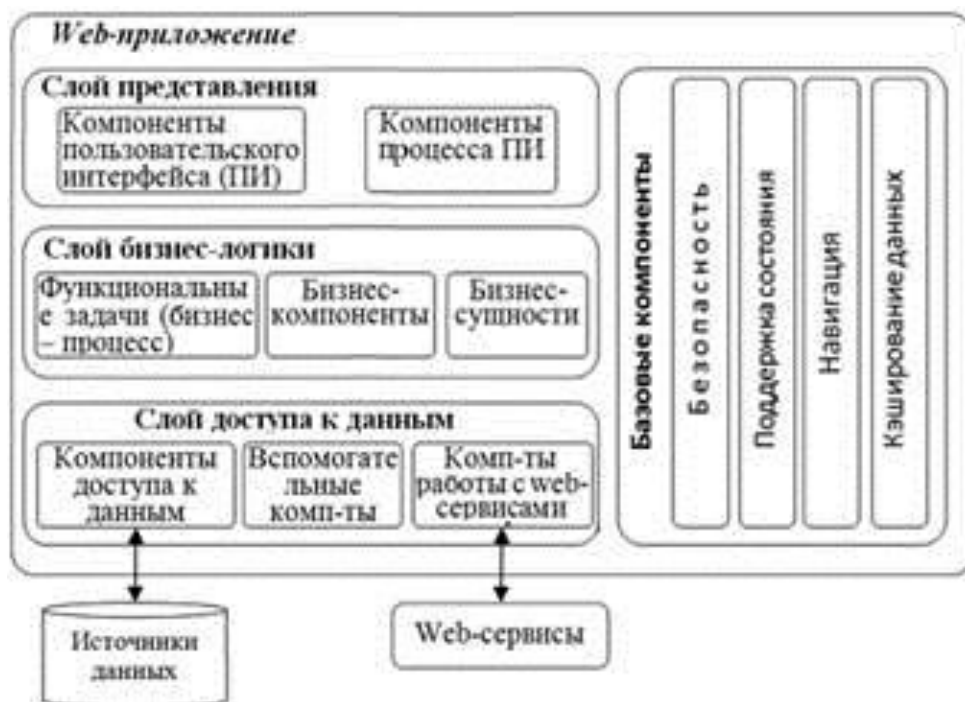


Рисунок 4.1 Типова структура web-додатку

Web-додаток також можна розділити на базові та функціональні підсистеми.

До базових підсистем web-додатки відносяться:

- оформлення web-сторінок, що становлять додаток;
- підтримка стану сеансу роботи користувачів;
- настройка web-сторінок для різних користувачів (персоналізація);
- навігація між різними web-сторінками;
- забезпечення безпеки (аутентифікація і авторизація, реєстрація користувачів);
- доступ до джерел даних.

До функціональних підсистем відносяться:

- управління контентом web-додатки (безліччю документів, зображень, посилань і т. д. "які доступні даному додатку). Web-додаток повинен надавати можливості завантаження контенту на сервер (uploading), класифікації, отримання з сервера (downloading);

- пошук інформації в контенті додатки (в документах, файлах даних, базах даних);

- підтримка взаємодії користувачів (форуми, обмін повідомленнями, рецензування документів);
- більш доступного режиму (розрахунки відповідно до бізнес-логікою, обробка зображень, обробка документів і т. д).

Базова функціональність багато в чому реалізується середовищем виконання web-додатки, а для створення функціональних підсистем треба скласти програмного коду.

Висновки до розділу 4

В даному розділі описано засоби, використані при розробці програмних додатків. Обґрунтовано вибір мови програмування та фреймворку (Java 8 EE, CSS, HTML, Vaadin Framework) та тому обрано відповідне середовище для роботи з мовою JAVA та JetBrains IntelliJ IDEA

5 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Донедавна єдиним можливим способом підключення електронних пристроїв до мережі Internet було використання спеціального комп'ютера (gateway computer), що займається трансляцією даних з Internet у внутрішню мережу керуючої системи відповідно до певного протоколу, а також, можливо, реалізує в собі частина функцій контролера.

Але на сьогоднішній день, у зв'язку з значним розвитком і великою популярністю мікроконтролерної техніки, стало можливим створення систем віддаленого керування й моніторингу без використання спеціальних комп'ютерів.

Переваги віддаленого моніторингу об'єктів через Internet:

керуюча система не прив'язана просторово до виконавчих пристроїв і датчиків, внаслідок чого з'являється можливість створення розподіленої системи моніторингу;

у більшості випадків відсутня необхідність прокладки додаткової кабельної мережі для зв'язку компонентів системи.

Керувати пристроями розумної системи можна різними способами, головне, щоб сигнал надійшов на сервер, а потім попрямував до контролера. Вибір способу управління і контролю залежить від особистих переваг, але зазвичай використовують:

- Настінні клавіші. Клавіші є запрограмованими пристроями, здатними посилати певний сигнал на сервер. За які саме дії будуть відповідати розумні клавіші, вирішує користувач. Це можуть бути цілі сценарії або поодинокі дії;
- Клавіші та дисплей. Web-інтерфейс зручний у використанні, на дисплеї миттєво відображається все обладнання і збережені бази даних.
- Ручка управління. Зазвичай роторна ручка є додатковим елементом до дисплея і розумним клавішах. Вона створена для більшої зручності.
- сенсорні кнопки відрізняються прекрасним дизайном і лаконічно вписуються в будь-який інтер'єр. Вони не перевантажені зайвими деталями, але відмінно справляються з поставленим завданням.

– Дистанційне керування через інтернет. Найзручніший спосіб для тих, хто постійно знаходиться поза домом. Контроль і управління розумними пристроями відбувається через web-інтерфейс, відображений на будь-якому мобільному пристрої через додаток. Всі необхідні дані будуть перебувати під рукою в будь-який час дня і ночі.

– Використання сценаріїв. Сценарії розумної системи надзвичайно зручні, так як дозволяють за секунди запустити кілька дій одночасно. Зазвичай сценаріями привласнюють відповідні назви, пов'язані з ситуацією, для якої будуть виконуватися дії. Сценарії економлять час, який раніше витрачався на рутину. Створювати нові сценарії і змінювати старі можна скільки завгодно раз.

Web-Інтерфейс стенду моделювання віддаленого моніторингу та управління технологічним процесом має такий вигляд (рисунки 5.1 — 5.4):



Рисунок 5.1 Стенд моделювання віддаленого моніторингу та управління технологічним процесом



Рисунок 5.2 Стенд моделювання віддаленого моніторингу та управління технологічним процесом (мікрокомп'ютер, мікроконтроллер, модулі)



Рисунок 5.3 Стенд моделювання віддаленого моніторингу та управління технологічним процесом



Рисунок 5.4 Стенд моделювання віддаленого моніторингу та управління технологічним процесом

На стенді моделювання віддаленого моніторингу та управління технологічним процесом ми бачимо Web-інтерфейс, який працює з мікрокомп'ютером.

Web-інтерфейс проводить моніторинг показників датчиків та елементи управління.

Мікрокомп'ютер ревізує web-інтерфейс та підтримує зв'язок зі стендом (рисунок 5.5).

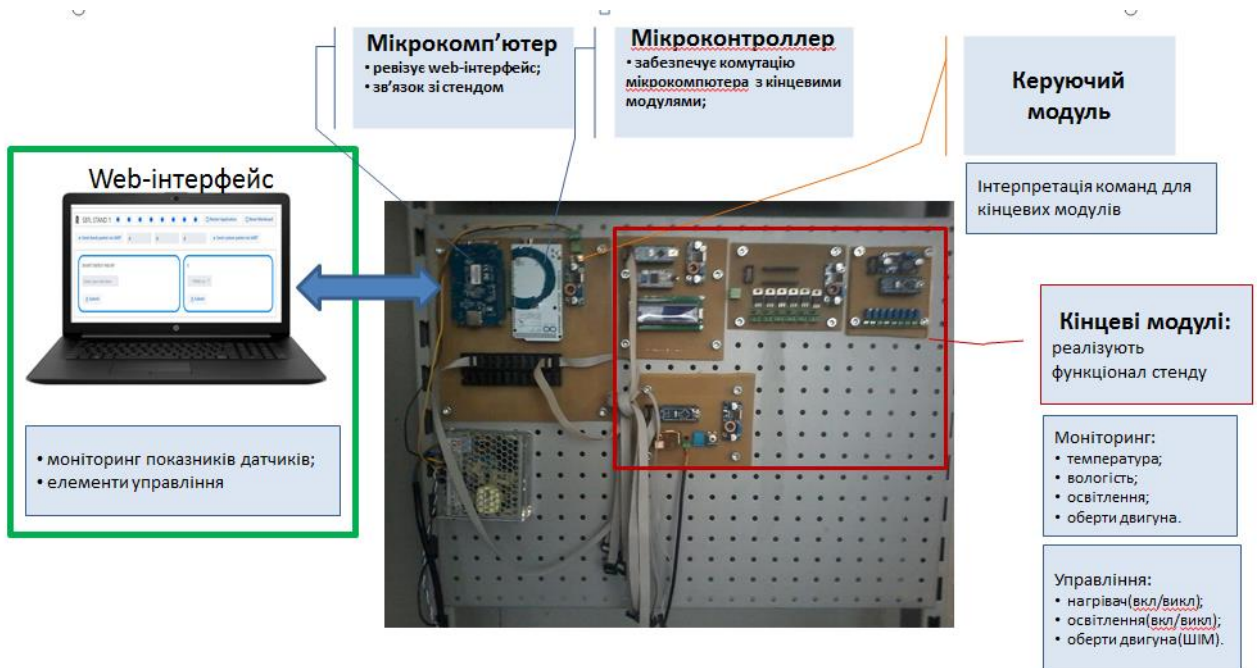


Рисунок 5.5. Опис стенду моделювання віддаленого моніторингу та управління технологічним процесом

На рисунку 5.6 представлені фрагменти коду клієнтської частини:

- Об'єкт повідомлення
- Сховище обробників різних отриманих повідомлень
- Фабрика створення об'єктів повідомлень з байткоду

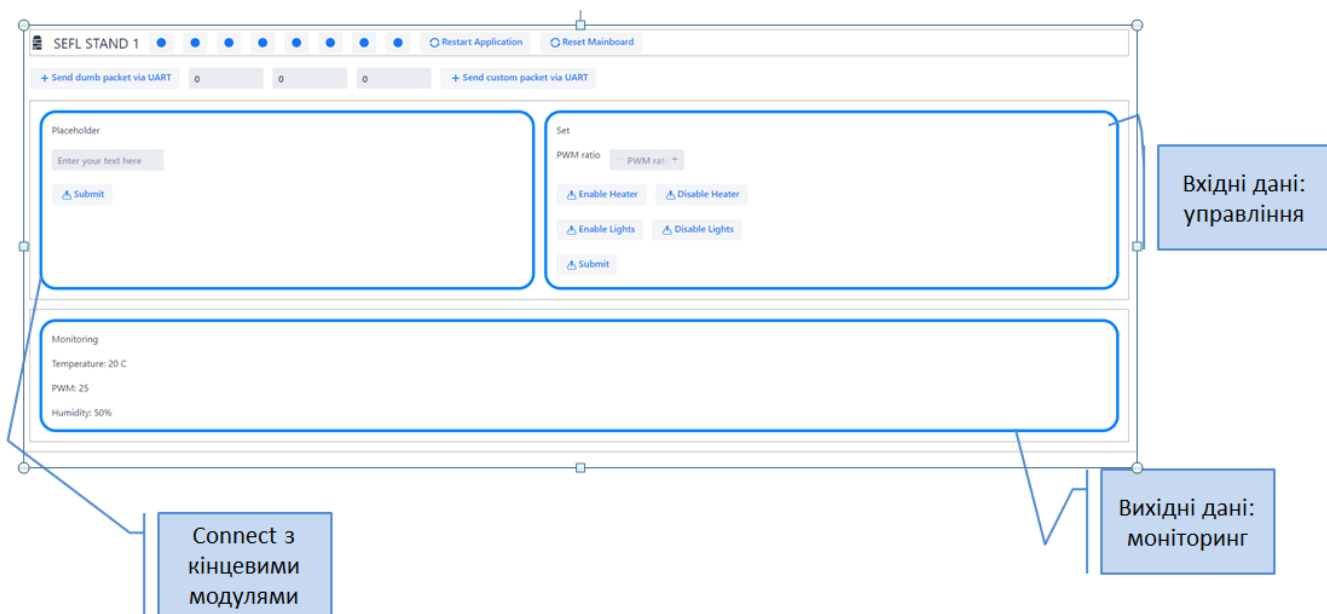


Рисунок 5.6 Інтерфейс користувача

Для початку роботи із системою клієнт через «Браузер» повинен завантажити «web-інтерфейс», після завантаження сторінки інтерфейсу запускається web-додаток. Потім встановлюється зв'язок з адаптерною платою, яка під'єднана до локальної системи керування фізичним об'єктом. Після встановлення зв'язку запускається функціональне ядро додатку, у якому періодично формуються спеціальні web-запити до адаптера й оновлюється інформація про стан об'єкту керування. Інформація може оновлюватися через певні проміжки часу або за запитом користувача.

При необхідності внести зміни до конфігурації адаптера або передати дані до об'єкту керування, web-додаток формує запит і відправляє його до адаптера. Той, в свою чергу, повинен відпрацювати запит і сформувавати звіт про результати виконання запиту й відправити назад до додатку. Отримані результати обробляються й відображаються клієнту на екрані. Використання такої структури програми й алгоритму роботи дозволяють завантажувати інтерфейс тільки один раз, що значно зменшує трафік, а отже й навантаження на мережу.

Програмне забезпечення мікроконтролера апаратного адаптера включає функціональне ядро, список властивостей, інтерфейси обміну даними та конфігурування ядра, низькорівневий драйвер обміну даними та USB-драйвер.

З точки зору споживача, характеристики, які вказані дозволяють отримати послуги з високим рівнем доступності і низькими ризиками непрацездатності, забезпечити швидке масштабування обчислювальної системи завдяки еластичності без необхідності створення, обслуговування і модернізації власної апаратної інфраструктури. Зручність і універсальність доступу забезпечується широкою доступністю послуг і підтримкою різного класу термінальних пристроїв (персональних комп'ютерів, мобільних телефонів, інтернет-планшетів).

Висновки до розділу 5

У даному розділі описано логіку роботи системи й подано інформацію для користувачів системи з метою полегшення їхньої подальшої взаємодії із розробленими додатками. Подано зразки інтерфейсу для спрощення сприйняття інструкції.

ВИСНОВКИ

Метою даного дипломного проекту було розроблення системи керування певним віддаленим об'єктом. Аналіз популярних систем керування, виконаний в роботі, показав доцільність та актуальність даної теми у промисловому та повсякденному житті.

Рішення для впровадження моніторингу та управління технологічним процесом запропоновано реалізувати, дотримуючись таких принципів: масштабованості, використання готових компонентів і базових проектних рішень, модульності, відкритості та сумісності.

Проаналізувавши існуючі системи віддаленого моніторингу об'єктів, можна прийти до висновку, що існуючі системи web-інтерфейсу вирішують певну локальну задачу та потребують у кожному випадку підлаштування під конкретний об'єкт чи обладнання. Деякі системи мають недоліки через велику вартість, інші не влаштовують щодо надання даних про технологічний об'єкт через спосіб реалізації або спосіб передачі даних. Таким чином, для максимального задоволення потреб клієнтів необхідно висунути наступні вимоги до максимально наближеної системи віддаленого моніторингу: висока швидкість обміну даними, низька вартість обслуговування, наявність web-інтерфейса, тобто відсутність прив'язки до одного ПК.

Для реалізації описаних вище функцій віддаленого моніторингу стану та керування обладнанням розроблено автоматизовану систему віддаленого моніторингу та управління технологічним процесом. У функції основи використовується програмне забезпечення розроблене у середовищі JetBrains IntelliJ IDEA.

Задіяні програми мають модульну структуру, що дозволяє оперативно допрацьовувати й модифікувати їх для конкретних ситуацій.

Апаратно-програмні компоненти повинні забезпечувати опрацювання даних у реальному часі з урахуванням обмежень щодо габаритів, енергоспоживання та вартості.

Основою системи моніторингу технологічних процесів є безпроводні сенсорні мережі, які забезпечують збір технологічних даних у реальному часі.

Була обрана дворівнева клієнт-серверна архітектура. Одна з основних її переваг – простота реалізації й можливість створювати розподілений доступ різним клієнтам з контролем повноважень і з різним рівнем доступу. Також це дозволить використовувати різні операційні системи та знизити вимоги до клієнтських ПК. Клієнтська частина — це, з одного боку, ПК із системою збору та передачі інформації, а з іншого — диспетчерський ПК з доступом до даних сервера по web-інтерфейсу.

Таким чином, була реалізована система віддаленого моніторингу та управління технологічним процесом, яка має власний web-інтерфейс, що дозволяє використовувати її з будь-якої точки планети, де присутнє з'єднання із всесвітньою мережею Інтернет. Грунтуючись на висунутих вимогах, була обрана дворівнева клієнт-серверна архітектура, яка має низку переваг перед іншими системами:

- високу швидкість обміну даними;
- низьку вартість обслуговування;
- наявність web-інтерфейсу;
- можливість прогнозування витрат на основі погодних умов;
- відсутність прив'язки до одного ПК;
- відсутність прив'язки до конкретної операційної системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Буч Г. Об'єктно-орієнтоване проектування з прикладами застосування / Г. Буч. — К.: Академія, 2002. — 723 с.
2. Взаємодія програмних систем [Електронний ресурс]. — Режим доступу до ресурсу: <https://geektimes.com/post/282922/>
3. Гриньов Д.В. Кодування зображень / Д.В. Гриньов. - Х.: НТУ „ХПІ”, 2003. - № 26. - (Вісник НТУ „ХПІ”: зб. наук. пр. Тематичний випуск: Інформатика і моделювання).
4. Євтушенко К.В., Перекрест А.Л. Обґрунтування структури мережевої лабораторії з дистанційним доступом через Internet // Збірка наукових праць VII Всеукраїнської науково-технічної конференції молодих учених і спеціалістів «Електромеханічні та енергетичні системи, методи моделювання та оптимізації». - Кременчук, КрНУ, 2010. - С. 55-56.
5. Медиковський М. О., Цмоць І. Г., Цимбал Ю. В. Інформаційноаналітична система для управління енергоефективністю підприємств Львова // Актуальні проблеми економіки: наук. екон. журнал. – № 1(175) 2014. – К., 2016. – С. 379–384.
6. Перекрест А.Л., Євтушенко К.В. Обґрунтування структури мережевої лабораторії з дистанційним доступом через Інтернет // Вісник КДУ імені Остроградського. – Вип. 4/2010(63). – Кременчук, 2010 – С. 183–187.
7. Перекрест А.Л., Найда В.В., Романенко С.С. та інш. Оперативний контроль температурних режимів і керування тепловими пунктами будівель навчального закладу// Вісник КрНУ. – Вип. 3/2013 (80). – Кременчук, 2013 – С. 35–43.
8. Проектування інформаційних систем: Посібник / За ред. В.С. Пономаренка. — К.: Академія, 2002. — 450 с.
9. Теслюк Т., Цмоць І., Опотяк Ю., Теслюк В. Архітектура багаторівневої системи управління енергоефективністю регіону // Вісник Нац. ун-ту

“Львівська політехніка” “Комп’ютерні науки та інформаційні технології”. – Львів, 2017. – № 864. – С. 201–209.

10. Хоріщенко Я.В., Перекрест А.Л. Система віддаленого керування та моніторингу технічних об’єктів з використанням Електромеханічні та енергетичні– Кременчук, 2012 – С. 49–51.
11. Шилдт Герберт - Java, повний посібник, 8-ме видання [Текст] : переклад з англ. - М. : ООО «І.Д. Вільямс», 2012.
12. Шипулін С. Тенденції розвитку ПЛІС і їх застосування для цифрової обробки сигналів / Шипулін С., Губанов Д., Стешенко В., Храпов В. - Електронні компоненти, 1999, №5. - с.42 - 45.
13. Цмоць І. Г., Теслюк Т. В., Машевська М. В., Теслюк В. М. Модель організації обміну та збереження даних у багаторівневих системах управління технологічними процесами // Науковий вісник Нац. лісотехн. ун-ту України: зб. наук.-техн. праць. – Львів: РВВ НЛТУ України. – 2017. – т. 27, № 1. – С. 197–202.
14. Цмоць І., Стрямець С. П., Зербіно Д. Д. Багаторівнева система управління технологічними процесами // Вісник Хмельницького нац. ун-ту, 2016. Технічні науки. – С. 139–145.
15. Цмоць І. Г., Скорохода О. В., Роман В. І. Сховища даних багаторівневих систем управління енергоефективністю // Моделювання та інформаційні технології: зб. наук. праць Інституту проблем моделювання в енергетиці. – 2016, Вип 77. – С. 192–197.
16. Цмоць І. Г., Теслюк Т. В., Машевська М. В., Теслюк В. М. Модель організації обміну та збереження даних у багаторівневих системах управління технологічними процесами // Науковий вісник Нац. лісотехн. ун-ту України: зб. наук.-техн. праць. – Львів: РВВ НЛТУ України. – 2017. – т. 27, № 1. – С. 197–202.
17. Teslyuk Taras, Tsmots Ivan, Teslyuk Vasyl, Medykovskyy Mykola and Opotyak Yuriy. Architecture and Models for System-Level Computer-Aided Design of the Management System of Energy Efficiency of Technological Processes at the

Enterprise // Advances in Intelligent Systems and Computing II, Advances in Intelligent Systems and Computing 689. Springer International Publishing AG 2018. P. 538–557.

18. Модулі RS232-Wi-Fi [Електронний ресурс]. – Режим доступу: <http://www.usconverters.com/serial-rs232-pci-card>.
19. Огляд NL6621 [Електронний ресурс]. – Режим доступу: <http://www.nufrontsoft.com/index.php/project/index/id/30.html>.
20. WiFi модуль RTL8710BN [Електронний ресурс]. – Режим доступу: <http://www.realtek.com/products/productsView.aspx?Langid=1&PFid=45&Level=4&Conn=3&ProdID=361>.
21. WiFi модуль ESP8266 [Електронний ресурс]. – Режим доступу: <https://esp8266.ru/>.
22. Одноплатний комп'ютер Raspberry Pi [Електронний ресурс]. – Режим доступу: <http://www.raspberrypi.org/archives/2180>.
23. BeagleBone Black [Електронний ресурс]. – Режим доступу: <https://beagleboard.org/black>.
24. Обчислювальна платформа від Intel [Електронний ресурс]. – Режим доступу: <https://software.intel.com/en-us/iot/hardware/edison>.
25. AllWinner pcDuino [Електронний ресурс]. – Режим доступу: <http://www.linksprite.com/linksprite-pcduino/>.
26. Cubieboard2 is here [Електронний ресурс]. – Режим доступу: <http://cubieboard.org/2013/06/19/cubieboard2-is-here/>.
27. OrangePi [Електронний ресурс]. – Режим доступу: <http://www.orangepi.org/>.
28. Проект Armbrian [Електронний ресурс]. – Режим доступу: <http://www.armbian.com/>.
29. Платформи для реалізації IoT рішень [Електронний ресурс]. – Режим доступу: <https://www.postscapes.com/internet-of-things-platforms/iot-device-connectivity-platform>.

ДОДАТОК А

Web-інтерфейс стенду моделювання віддаленого моніторингу та
управління технологічним процесом

Специфікація

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ТВз51

Аркушів 2

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ"КПР"_ТЕФ_АПЕПС_ТВз51	Записка.docx	Текстова частина дипломної роботи
Компоненти		
УКР.НТУУ"КПР"_ТЕФ_АПЕПС_ТВз51	Об'єкт повідомлення	Основний компонент додатку
УКР.НТУУ"КПР"_ТЕФ_АПЕПС_ТВз51	Сховище обробників різних отриманих повідомлень	Основний компонент додатку
УКР.НТУУ"КПР"_ТЕФ_АПЕПС_ТВз51	Фабрика створення об'єктів повідомлень з байткоду	Основний компонент додатку

ДОДАТОК Б

Web-інтерфейс стенду моделювання віддаленого моніторингу та управління технологічним процесом

Текст програми

УКР.НТУУ"КП" _ТЕФ_АПЕПС_ТВз51

Аркушів 2

Київ 2019

ТЕКСТ ПРОГРАМИ – ОБ’ЄКТ ПОВІДОМЛЕННЯ

```
public class MainboardDevice extends Device implements MainboardCommands {
    private byte currentStatus;
    private MainboardListeners mainboardListeners;
    public MainboardDevice(byte id) {
        super(id);
        setType(DeviceConstants.MAINBOARD_TYPE_ID);
        mainboardListeners = MainboardListeners.getInstance();
    }

    public byte getCurrentStatus() { return currentStatus; }

    @Override
    public void parseCommand(byte[] command) {
        switch (command[0]){
            case STATUS_COMMAND: receiveStatusCommand(command);
                break;
            default:
        }
    }

    @Override
    public void onDataReceived(DataMessage dataMessage) {
        switch (dataMessage.getCommandId()){
            case STATUS_COMMAND:
                receiveStatusCommand(dataMessage.getCommandData());
                break;
            default:
        }
    }

    public void receiveStatusCommand(byte[] data) {
        this.currentStatus = data[1];

        new Thread()->{
            for(StatusListener listener: mainboardListeners.getStatusListeners())
                new Thread()->listener.receive(currentStatus);
        }.start();
    }
}
```

ДОДАТОК В

Web-інтерфейс стенду моделювання віддаленого моніторингу та управління технологічним процесом

Текст програми

УКР.НТУУ"КП" _ТЕФ_АПЕПС_ТВз51

Аркушів 2

Київ 2019

ТЕКСТ ПРОГРАМИ – СХОВИЩЕ ОБРОБНИКІВ РІЗНИХ ОТРИМАНИХ ПОВІДОМЛЕНЬ

```
public class MessageFactory {  
    public static DataMessage getDataMessage(byte[] message){  
        DataMessage result = new DataMessage();  
        result.setMessage(message);  
        return result;  
    }  
  
    public static DataMessage getDataMessageWithEND(byte[] message){  
        DataMessage result = new DataMessage();  
        byte[] actualMessage = new byte[message.length+1];  
        actualMessage[actualMessage.length-1] = 0;  
        System.arraycopy(message, srcPos: 0, actualMessage, destPos: 0, message.length);  
        result.setMessage(actualMessage);  
        return result;  
    }  
  
    public static Message getMessage(byte[] message) { return getDataMessage(message); }  
}
```

ДОДАТОК Г

Web-інтерфейс стенду моделювання віддаленого моніторингу та управління технологічним процесом

Текст програми

УКР.НТУУ"КП" _ТЕФ_АПЕПС_ТВз51

Аркушів 2

Київ 2019

ТЕКСТ ПРОГРАМИ – ФАБРИКА СТВОРЕННЯ ОБ’ЄКТІВ ПОВІДОМЛЕНЬ З БАЙТКОДУ

```
public class DataMessage extends Message {
    @Override
    public String toString() {
        return "DataMessage{" +
            "message=" + Arrays.toString(message) +
            ", valid=" + valid +
            '}';
    }

    public void setMessage(byte[] message) {
        this.message = message;
        if(isValid()) valid = true;
        else valid = false;
    }

    public byte getTarget() {
        if(isValid()) return message[0];
        return 0;
    }

    public byte[] getCommandAndParameters() {
        if(isValid()) {
            byte[] params = new byte[message.length-2];
            System.arraycopy(message, srcPos: 1, params, destPos: 0, length: message.length-2);
            return params;
        }
        return null;
    }

    public byte getCommandId() {
        if(isValid()) {
            return message[1];
        }
        return -1;
    }

    public byte[] getCommandData() {
        if(isValid()) {
            byte[] params = new byte[message.length-3];
            System.arraycopy(message, srcPos: 2, params, destPos: 0, length: message.length-3);
            return params;
        }
        return null;
    }
}
```

ДОДАТОК Д

Web-інтерфейс стенду моделювання віддаленого моніторингу та
управління технологічним процесом

Опис програми

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТВ351

Аркушів 9

Київ 2018

АНОТАЦІЯ

Даний додаток містить опис трьох програмних систем розроблених для створення web-інтерфейсу стенду моделювання віддаленого моніторингу та управління технологічним процесом. Створені програмні системи показують об'єкт повідомлення, сховище обробників різних отриманих повідомлень та фабрику створення об'єктів повідомлень з байткоду та виконують такі завдання:

- введення запиту;
- конвертація даних в необхідний для системи вигляд;
- збір даних;
- виведення результатів.

Програми отримують дані через елементи керування.

При розробці цих програмних систем використовувалась мова JAVA та інші допоміжні мови та технології: EJB, Vaadin Framework, SEFL SDK (Configuration, UART), JSSC, junit, JetBrains IntelliJ IDEA, Apache Tomcat.

ЗМІСТ

1. Загальні відомості.....	65
2. Функціональне призначення	66
3. Опис логічної структури.....	67
4. Технічні засоби, що використовуються	68
5. Виклик і завантаження.....	69
6. Вхідні і вихідні дані	70

ЗАГАЛЬНІ ВІДОМОСТІ

У цьому додатку міститься опис трьох програмних систем. У додатку Б міститься програмний код головних модулів розроблених програмних систем.

Обидва розроблені додатки працюють в операційній системі Windows, Android та інші системи на яких є підключення до інтернету.

При розробці обох програмних додатків використовувалась мова JAVA та з використанням двох середовищ:

1. Інтерфейс користувача.
2. Мікрокомп'ютер.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Однією з найважливіших задач, що виникають при розробці і впровадженні автоматизованих систем та автоматики керування технологічними об'єктами, є забезпечення безперервного моніторингу, контролю та оперативного віддаленого керування технологічним процесом.

Розроблені компоненти виконують такі завдання: зрозумілий інтерфейс та зручна система навігації; надійність та неможливість керування сервером третіми особами; можливість кастомізувати інтерфейс.

Також розроблені додатки можуть використовуватися в якості учбових матеріалів при опрацюванні технологій віддаленого моніторингу та управління технологічним процесом.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Для створення та реалізації web-інтерфейсу стенду моделювання віддаленого моніторингу та управління технологічним процесом були використані такі технології: Java 8 EE, EJB, Vaadin Framework, SEFL SDK (Configuration, UART), JSSC, junit, JetBrains Intel, IDEA, Apache Tomcat.

При запуску систем спочатку з'являється вікно форми для введення даних, формується запит і відправляє його до адаптера. Той, в свою чергу, повинен відпрацювати запит і сформувати звіт про результати виконання запиту й відправити назад до додатку. Отримані результати обробляються й відображаються клієнту на екрані.

ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для забезпечення повноцінної роботи та досягнення високої ефективності роботи створених додатків для демонстрації роботи web-інтерфейсу стенду моделювання віддаленого моніторингу та управління технологічним процесом використовувались Java, HTML, CSS та Vaadin Framework, які показали себе як надійна та гнучка середовище розробки програм.

Розроблені додатки працюють в операційних системах Windows, Android та інших системах на яких є підключення до інтернету.

ВИКЛИК І ЗАВАНТАЖЕННЯ

Розроблені системи не потребують інсталяції, достатньо запустити виконуваний файл.

Після запуску користувач отримає доступ до головної форми програми, звідки може виконувати необхідні обчислення.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними для розроблених додатків є запит, який зчитується з `textBox`.

Вхідні дані можуть мати вигляд тексту або коду.

Вихідними даними є також текст або код.

Вихідними даними є результуюче значення температури та вологості.

ДОДАТОК Е

Web-інтерфейс стенду моделювання віддаленого моніторингу та управління технологічним процесом

Акт впровадження

УКР.НТУУ"КПІ" _ТЕФ_АПЕПС_ТВ351

Аркушів 2

Київ 2019