

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

**Інститут інформатики та радіоелектроніки**  
**Факультет комп'ютерних наук та технологій**  
(повне найменування інституту, факультету)

**Кафедра комп'ютерних систем та мереж**  
(повне найменування кафедри)

## Пояснювальна записка

до дипломного проекту (роботи)

бакалавра

(ступінь вищої освіти (освітній ступінь))

на тему **РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ ДЛЯ РОЗУМНОГО БУДИНКУ**  
**З ВИКОРИСТАННЯМ ВЕБ ТЕХНОЛОГІЙ**

Виконав: студент 4 курсу, групи КНТ-517  
спеціальності \_\_\_\_\_

123 «Комп'ютерна інженерія»

(код і найменування спеціальності)

Освітня програма (спеціалізація) \_\_\_\_\_

«Комп'ютерна інженерія»

Шеховцова Тетяна Андріївна

(прізвище та ініціали)

Керівник Ільяшенко М.Б.

(прізвище та ініціали)

Рецензент Куляба-Харитонова Т.І.

(прізвище та ініціали)

м. Запоріжжя  
2021 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**Національний університет «Запорізька політехніка»**  
(повне найменування вищого навчального закладу)

Інститут, факультет інформатики та радіоелектроніки, комп'ютерних наук і технологій  
Кафедра «Комп'ютерні системи та мережі»  
Ступінь вищої освіти (освітній ступінь) бакалаврський  
Спеціальність 123 Комп'ютерна інженерія  
(код і найменування)  
Освітня програма (спеціалізація) Комп'ютерна інженерія  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри Кудерметов Р.К.

“ ” 2021 року

**ЗАВДАННЯ**  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТА

Шеховцова Тетяна Андріївна

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка системи управління для розумного будинку з використанням веб технологій

керівник проекту (роботи) Ільяшенко Матвій Борисович, к. т. н., доцент

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

затверджені наказом вищого навчального закладу від “17” березня 2021 року № 81

2. Строк подання студентом проекту (роботи) 06 травня 2021 року

3. Вихідні дані до проекту (роботи) голосові команди користувачей, температура та вологість у приміщеннях.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Вступ; дослідження предметної області використання боту; аналіз існуючих засобів та вибір найкращих варіантів для системи моніторингу і диспетчеризації; розробка Telegram боту для моніторингу поточного стану серверної кімнати


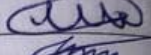


5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

UML-діаграма послідовності;

UML-діаграма прецедентів;

UML-діаграма діяльності клієнта;

Результати роботи.

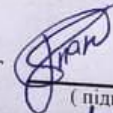
6. Консультанти розділів проекту (роботи)		Підпис, дата	
Розділ	Прізвище, ініціали та посада консультанта	завдання видав	приймав виконане завдання
1-3	Ільяшенко М.Б., к. т. н., доцент		
НК	Щербак Н.В. ст. викл		

7. Дата видачі завдання 01.03.2021 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Варіантний огляд та аналіз сучасної системи управління «розумний дім»	01.03.2021 р.	
2	Аналіз засобів систем управління «Розумного» будинку	15.03.2021 р.	
3	Розробка архітектури системи	20.03.2021 р.	
4	Розробка програмного забезпечення системи	01.04.2021 р.	
5	Розрахунок технічних характеристик системи	10.04.2021 р.	
6	Розрахунок економічної доцільності розробки системи	15.04.2021 р.	
7	Розробка UML-діаграм системи	20.04.2021 р.	
8	Тестування програмного забезпечення	25.04.2021 р.	
9	Захист ДП	25.05.2021 р.	

Студент



(підпис)

Керівник проекту (роботи)



(підпис)

Т.А. Шеховцова  
(ініціали та прізвище)

М.Б.Ільяшенко  
(ініціали та прізвище)

## РЕФЕРАТ

ПЗ: 81 с., 38 рис., 4 табл., , 89 посилань, 3 додатка.

### РОЗУМНИЙ ДІМ, ЗБІР ІНФОРМАЦІЇ, ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ, ЗБЕРІГАННЯ ІНФОРМАЦІЇ, КОНТРОЛЕР

В ході виконання дипломної роботи бакалавра було проаналізовано різні види розумних будинків та розроблено систему, за допомоги якої житель будинку зможе регулювати температуру, спостерігати за станом будинку та керувати їм за допомоги голосового управління. Програмні засоби мають зручний та зрозумілий інтерфейс для користувача.

В першому розділі був проведений аналіз об'єкту автоматизації, розглянуті сучасні комп'ютеризовані системи управління розумним будинком. Зроблена постановка задачі.

В другому розділі для даної системи було проведено аналіз функцій, та на його основі побудована UML-діаграма прецедентів. Було розроблено структурну та функціональну схему даної системи. Було розроблено UML-діаграми діяльності та послідовності.

В третьому розділі було проаналізовано та підібрані механічні засоби для роботи з системою, розроблено програмне забезпечення для клієнта, веб-сервісу та мікроконтролеру на базі плати Arduino.

Після проведення тестування можна зробити висновок, що розроблена система може бути використана на сучасних розумних будинках.

Було розроблено структурні схеми та UML-діаграма діяльності, UML-діаграма прецедентів. Також розроблено схему обладнання, яке використовується у системі.

## ЗМІСТ

Вступ.....	7
1 Аналіз сучасних систем управління "Розумний дім" .....	9
1.1 Застосування та можливості систем управління «Розумний дім» .....	9
1.2 Аналіз систем управління «Розумний дім» .....	13
1.2.1 Огляд Amazon Echo .....	13
1.2.2 Огляд Nest Learning Thermostat.....	14
1.2.3 Огляд Apple HomeKit Netatmo .....	15
1.2.4 Огляд Meizu Lifekit .....	16
1.2.5 Огляд Google Home .....	18
1.2.6 Висновки оглядів сучасних систем управління для розумних будинків.....	20
1.3 Проблеми автоматичного розпізнавання мови .....	20
1.4 Варіантний аналіз і обґрунтування вибору програмних засобів для реалізації веб-системи .....	24
1.4.1 Мова C # та технологія ASP. NET Core .....	25
1.4.2 Платформа Java EE та технологія Java Persistence API .....	25
1.4.3 PHP-скрипти .....	26
1.4.4 JavaScript та об'єкт XMLHttpRequest .....	27
1.5 Вибір напрямку та задач дослідження .....	30
2 Проектування голосової системи управління «Розумний дім» .....	31
2.1 Аналіз функцій комп'ютеризованої системи управління «Розумний дім».....	31
2.2 Розробка структурної схеми КСУ розумним домом .....	34
2.3 Розробка схеми автоматизації КСУ розумним будинком.....	34
2.4 Дослідження та удосконалення алгоритму розпізнавання голосу .....	35
2.5 Розробка UML-діаграм функціонування КСУ розумним домом .....	39
3 Розробка системи управління «Розумний дім».....	43
3.1 Обґрунтування технічних засобів КСУ розумним домом .....	43
3.2 Розробка клієнта з удосконаленою бібліотекою для системи «Розумний дім».....	49

Висновки .....	6 57
Перелік джерел посилання .....	58
Перелік графічного матеріалу:	
Пл.1 – UML-діаграма послідовності	
Пл.2 – UML-діаграма прецедентів	
Пл.3 – UML-діаграма діяльності клієнта	
Пл.4 - Результати роботи	

## ВСТУП

Актуальність: системи «розумний дім» швидко розвиваються, вони використовуються для забезпечення комфорту, зручності, якості життя та безпеки для мешканців. В даний час більшість систем, використовуються для зменшення людської праці [1-2]. Система «розумний дім» може бути спроектована та розроблена за допомоги одного контролера, який має можливість керувати та контролювати різні пов'язані між собою пристрої, такі як штепсельні вилки, світильники, датчики температури та вологості, димові, газові та пожежні сповіщувачі, а також аварійні та охоронні системи [3]. Одна з найбільших переваг системи полягає в тому, що можна легко керувати та управляти масивом пристроїв, таких як смартфон, планшет, настільний комп'ютер та ноутбук [4]. На даний час розробка систем «розумний дім» є дуже актуальними по всьому світу .

Мета: удосконалення швидкодії розпізнавання голосових команд існуючої бібліотеки, за рахунок використання словників та асинхронних методів програмування та впровадження її в комп'ютеризовану систему «Розумний дім», яка повинна бути розроблена за сучасною трьохрівневою архітектурою.

Об'єкт дослідження: об'єктом дослідження є удосконалення не ресурсномісткого алгоритму розпізнавання мови та застосування його на клієнті, який в свою чергу буде підключений до серверу у систему «розумний дім».

Предмет дослідження: методи розпізнавання голосових команд, розробка клієнта, сервера та розробка системи «Розумний дім» на базі мікроконтролера Arduino Uno.

Завдання: завдання даної роботи полягає в удосконаленні існуючого алгоритму для розпізнавання голосових команд та в розробці практичної частини, а саме реалізації клієнта, сервера та розробки системи «Розумний дім» у вигляді лабораторного стенду.

Наукова новизна: удосконалений алгоритм розпізнавання голосових команд дає змогу зменшити час розпізнавання, за рахунок використання асинхронних методів та додаткових довідників при розробці клієнта.

Практична цінність: розроблена система та практична реалізація дає змогу використовувати для впровадження в сучасні будинки та робити життя власників безпечнішим, та легким в управлінні. Розроблена система може також буду використана при розробці інших клієнтів, які потребують у розпізнаванні голосових команд.



# 1 АНАЛІЗ СУЧАСНИХ СИСТЕМ УПРАВЛІННЯ "РОЗУМНИЙ ДІМ"

## 1.1 Застосування та можливості систем управління «Розумний дім»

Розумний дім – це інтеграція та управління освітленням, безпекою, мультимедіа, клімат-контроль та іншими електронними системи в межах будинку. Зазвичай, це підмножина автоматизації в будівлі, яка зосереджується на житлових приміщеннях у масштабі квартири або сімейного будинку. Головна мета – зробити повсякденне життя більш комфортне, безпечне та енергоефективне [5].

Пристрої розумного будинку взаємо з'єднуються в мережу та контролюються спеціалізованим програмним забезпеченням. Яке підключається до пристроїв, контролює їх стан і реагує на події. Система виконує багато завдань, що дозволяє власникам будівлі витратити більше часу на питання, які дійсно важливіші. Система управління не тільки забезпечує комфортність, зберігає повітря свіжим та температуру комфортною, контролює опаленням та вентиляцію, а також дозволяють здійснювати автоматизовані операції управління:

- освітленням;
- безпекою;
- побутовими приборами;
- водопостачанням та газопостачанням [6].

Усі перелічені операції, можуть контролюватися, як у будинку або у віддаленому доступі (рис.1.1).

Наприклад, існує можливість відкривати та закривати двері та вікна будинку або вимикати всі вогні однією кнопкою у вашій спальні. Прокидатися під улюблені мелодії, та з кип'яченою водою у чайнику для готування кави, або чаю. Коли ви виїжджаєте з гаражу, брама відчиняється одною голосовою командою. Після виїзду, та промови голосової команди на закриття дверей, дверні замки автоматично зачиняються та система активується система безпеки. Розумний дім може запускати полив газону, відповідно до прогнозу погоди з Інтернету [7].

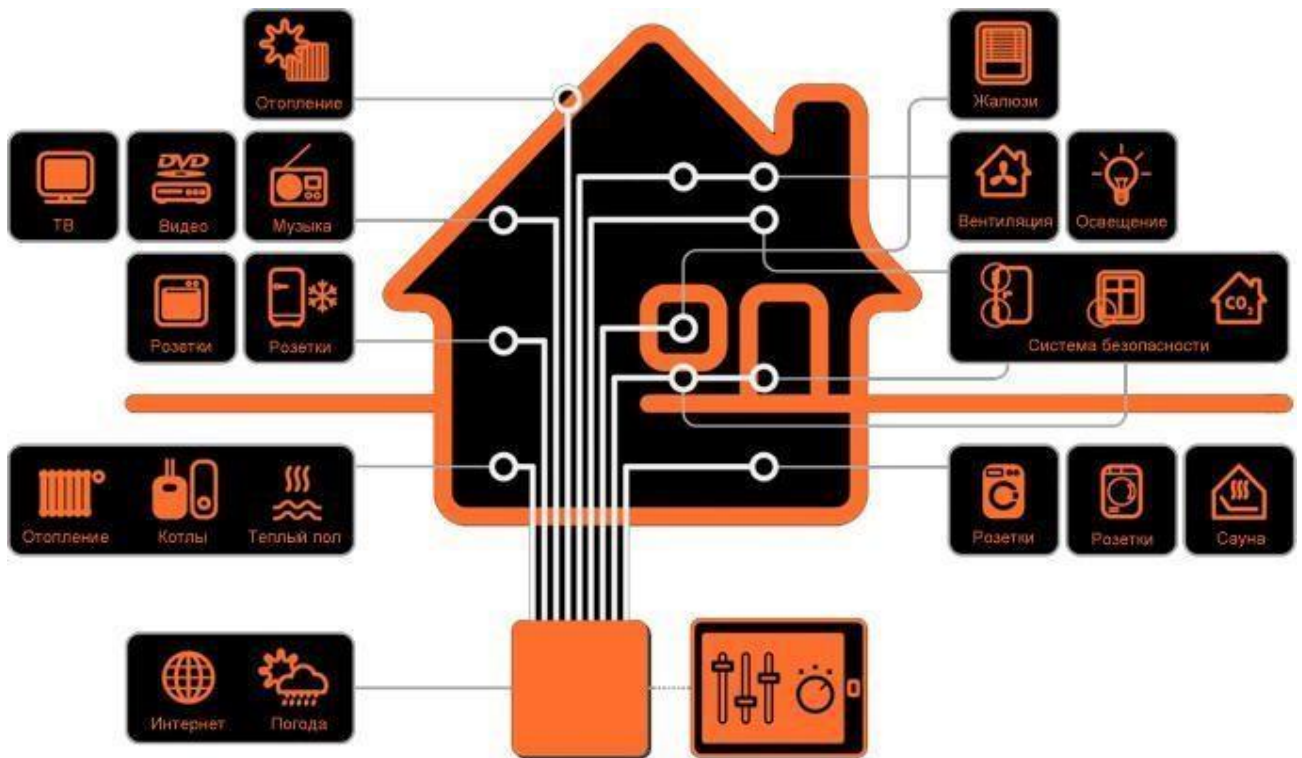


Рисунок 1.1 – Приклад системи «Розумний дім»

Незважаючи на те, що ця ідея звучить дуже технічно та футуристично, справжній розумний дім інтуїтивно зрозумілий і простий у використанні, так що кожен, хто живе у родині, має змогу використовувати його можливості на всі сто відсотків. У майбутньому розумні будинки, швидше за все, стане такими ж поширеними, як пральні машини. На сьогоднішній день кожен другий новий будинок в США має вбудовану систему [8].

Голосові помічники, такі як Amazon Alexa (рис. 1.2) та Google Home (рис. 1.3) [9], як і раніше, відчуються нам дивними, але вони популярні в основному через те, наскільки зручні та прості у використанні. Саме з цієї причини вони є однією з перших розумних рішень, які споживачі додають до своїх будинків. Виробники сучасних пристроїв для домашньої автоматизації знають це і швидко беруть на борт, щоб їх електронні та освітлювальні прилади, могли контролюватися за допомогою голосових помічників. Кухня є, найчастіше використаною зоною, для голосових помічників, це дає людям можливість робити вільні руки [9].



Рисунок 1.2 – Голосовий помічник від Amazon «Alexa»

Автоматизація для кухні, можуть зняти питання про вимірювання з домогосподарок, встановити таймер, а також вказувати список дії та необхідні інгредієнти для приготування улюблених страв. А голосовий контроль - це загальне зручність, де він навіть може бути інструментом для організації чистоти у оселі; замість того, щоб натискати на вимикач світла брудними руками, домовласники можуть надати голосові команди для помічника, який в свою чергу, виконає задані дії. Але, будучи в змозі включити світло тут, і там це не зовсім розумний будинок, просто контроль і зручність. Використання голосових команд дійсно починає стати корисним, коли ми надаємо голосовим помічникам можливість робити справжню роботу. Система домашнього інтелектуального дому цілком вирішує цю проблему однією командою, яка може ініціювати дії з кількох пристроїв одночасно. Коли є освітлення, телебачення, динаміки та клімат-контроль пов'язаний разом як одне ціле, голосовий помічник стає набагато корисніше. Очевидно одне: споживачі чекають голосовий контроль для всієї оселі у будівлі [10].



Рисунок 1.3 – Голосовий помічник Google Home

На сьогоднішній день ринок розумних будинків став досить великим, щоб мати спеціалізовані рішення. Виникла велика кількість невеликих постачальників домашньої автоматизації, а також гіганти автоматизації почали усвідомлювати необхідність рішень, призначених для домашнього використання [11]. Проте пропонувані системи часто дорогі, складні та негнучкі. З іншого боку, дійсно прості і доступні рішення, такі як X10 [12], все ще мають дуже обмежену функціональність. Ось чому ринок домашньої автоматизації значно відстає від очікувань та можливостей. У нас вже є всі технології та обладнання, щоб

побудувати комерційно успішні Розумні будинки, про які сьогодні тільки мріють люди. Це все просто справа хорошого програмного забезпечення [13].

## **1.2 Аналіз систем управління «Розумний дім»**

В наш час існує чимало відомих та нових брендів, але, зазвичай, в усіх є переваги та недоліки. Саме цьому і присвячено цей підрозділ, в якому оглянемо та визначмо сильні та слабкі сторони світових брендів, серед яких:

- Amazon Echo;
- Nest Learning Thermostat;
- Apple HomeKit Netatmo;
- Meizu LifeKit;
- Google Home [14].

Головні параметрами в нас будуть наступні:

- масштабованість;
- ціна;
- підтримка операційних систем сучасних смартфонів;
- мова сприйняття команд, якщо це передбачено;
- підтримка сторонніх засобів управління [15].

### **1.2.1 Огляд Amazon Echo**

Amazon Echo (скорочено називається Echo) – це марка розумних носіїв, розроблених Amazon.com [16]. Апарати підключаються до голосового інтелектуального персонального голосового помічника Alexa, який відповідає на ім'я "Alexa". Це слово пробудження, може змінюватися користувачем на "Amazon", "Echo" або "Computer". Пристрій підтримує голосові взаємодії, відтворення музики, створення списків справ, налаштування нагадувань, трансляцію подкастів, відтворення аудіокниг, інформацію про погоду, трафіку та

іншої інформації в реальному часі. Він також може керувати кількома розумними пристроями, використовуючи себе як концентратор домашньої автоматизації [17].

Має наступний ряд переваг:

- вбудований голосовий помічник;
- легко масштабується;
- відносна невелика ціна;
- підтримка усіх сучасних смартфонів;
- має можливість під'єднати сторонні засоби домашньої автоматизації [19].

Недоліки:

- не сприймає українську та російську мову [20].

### 1.2.2 Огляд Nest Learning Thermostat

Nest Learning Thermostat – це розумний термостат, розроблений Nest Labs (рисунок 1.4). Це електронний, програмований пристрій, який здатен до навчання. Пристрій із підтримкою Wi-Fi, який оптимізує опалення та охолодження будинків та підприємств для економії енергії [21].



Рисунок 1.4 – Вигляд системи Nest Learning Thermostat

Заснована на алгоритмі машинного навчання: протягом перших тижнів користувачі повинні регулювати термостат, щоб забезпечити набір еталонних

даних. Після чого Nest Learning Thermostat навчиться графіку людей, яку температурі вони використовують і коли [22]. Використовуючи вбудовані, він може перейти в режим енергозбереження, коли він усвідомлює, що ніхто не вдома.

Переваги:

- використання алгоритму машинного навчання;
- мала ціна;
- ручне керування через Wi-Fi [23].

Недоліки:

- не масштабується;
- не підтримує сторонні засоби управління;
- малий функціонал;
- відсутність вбудованої системи розпізнавання голосових команд [24].

### **1.2.3 Огляд Apple HomeKit Netatmo**

HomeKit - це програмне забезпечення компанії Apple, яке дозволяє користувачам налаштувати свій iPhone або іншій пристрій Apple для керування смарт-побутовою технікою, має наступний вигляд – рисунок 1.5. Користувачі можуть вмикати електроприлади в будинку за допомогою простого диктування голосом до Siri або через додатки [25]. HomeKit був вперше випущений з iOS 8 у вересні 2014 року.

Виробники пристроїв з підтримкою HomeKit повинні придбати програму MFI [26], і всі продукти HomeKit повинні мати копроцесор шифрування [27]. Обладнання, вироблене без підтримки HomeKit, може бути увімкнено для використання через продукт "шлюз", наприклад, концентратор, який з'єднує ці пристрої та службу HomeKit [28]:

- вбудований голосовий помічник;
- підтримка усіх сучасних смартфонів;
- має можливість під'єднати сторонні засоби домашньої автоматизації;
- багатий перелік готових функцій [29].

Недоліки:

- велика ціна;
- є проблеми з масштабованістю;
- не підтримує OS Android [30].



Рисунок 1.5 – Вигляд системи управління Apple HomeKit Netatmo

#### 1.2.4 Огляд Meizu Lifekit

Китайська компанія Meizu також долучилася до виробників-гігантів домашньої автоматизації, випустивши нову системою Lifekit, яка складається з декількох пристроїв, включаючи розумну лампочку (X-Plus SMART Bulb), підлогові ваги (RuFit), очищувач повітря (Air Box), розумний кондиціонер (Smart A/C), який контролює якість повітря у домі [31], усі пристрої зображено на рисунках 1.6-1.9.

Ці продукти розроблені у партнерстві з Haier і Alibaba, в основному однаковим устаткуванням, але вони інтегровані в платформу Lifekit Meizu. Meizu, Haier і Alibaba будуть співпрацювати, щоб запропонувати додаткові пристрої, які



будуть інтегровані в Lifekit, і запропонують SDK, що дозволить стороннім додаткам приєднатися [32].



X-Light Plus  
smart lamp

Рисунок 1.6 – Розумна світлодіодна лампочка X-Plus SMART Bulb



RyFit  
smart scale

Рисунок 1.7 – Підлогові ваги RyFit

Управління кожним з перерахованих вище пристроїв здійснюється за допомогою Android-смартфона, на якому попередньо повинно бути встановлено спеціальний додаток Life Kit. Система відноситься до змішаного типу [33].



Air Box

Рисунок 1.8 – Очищувачі повітря Air Box



Рисунок 1.9 – Розумний кондиціонер Smart A/C

Переваги:

- наявність готових пристроїв;
- простота установки і налаштування;
- функції керуючого пристрою бере на себе телефон [34].

Недоліки:

- порівняно малий функціонал;
- обмежений вибір устаткування;
- прив'язка до пристроїв від Meizu [35].

### 1.2.5 Огляд Google Home

Система від Google дозволяють користувачам використовувати команди для взаємодії з сервісами через інтелектуального персонального помічника Google, який називається Google Assistant [36], вона зображена на рисунку 1.10. Велика кількість функцій, як внутрішніх, так і сторонніх, інтегровано в систему. Це дозволяє користувачам слухати музику, керувати відтворенням відео та фотографій або отримувати новини і все це за допомоги голосових команд. Пристрої Google також мають вбудовану підтримку домашньої автоматизації, дозволяючи користувачам керувати розумними побутовими приладами своїм голосом. Домашні пристрої від Google можна розміщувати в різних кімнатах у

будинку, для синхронного відтворення музики. Оновлення в квітні 2017 року забезпечило підтримку декількох користувачів, що дозволило пристрою розрізняти до шести осіб голосом [37]. У травні 2017 року Google оприлюднила декілька оновлень для функціональності Google Home, зокрема: безкоштовні телефонні дзвінки в режимі "вільні руки" в США та Канаді; оновлення випереджають заплановані події [38].



Рисунок 1.10 – Вигляд системи Google Home

Переваги:

- вбудований голосовий помічник;
- масштабованість;
- підтримка усіх сучасних смартфонів;
- має можливість під'єднати сторонні засоби домашньої автоматизації;
- велика кількість мов для розпізнавання;
- великий вибір готових функції [39].

Недоліки:

- висока ціна;
- повільна швидкість розпізнавання [40];

### **1.2.6 Висновки оглядів сучасних систем управління для розумних будинків**

Отже, провівши огляд, було виявлено, перелік проблем, які є наданий час у сучасних системах управління для розумних будинків. Зазвичай це висока ціна, малий перелік готових функцій, відсутність україномовного та російськомовного розпізнавання голосових команд. Слід відзначити, що сучасні бренди поступово намагаються вирішити їх, але це складне завдання [41].

## **1.3 Проблеми автоматичного розпізнавання мови**

Як і більшість систем, наша система, буде мати голосове управління, тому розглянемо питання розпізнавання мови.

Дослідження, що стосуються області розпізнавання мови, визначаються двома основними сферами: фундаментальними дослідженнями, цілі яких спрямовані на проведення розробки і тестування нових методик, алгоритмів і підходів на некомерційній основі; і відповідних досліджень, які пов'язані з тим, щоб поліпшувати існуючі способи, орієнтуючись на певні критерії [42].

Для систем з розпізнавання мови, які мають слова, процеси розпізнавання визначають проведення порівнянь серед вхідними словами і тими словами, які є в словнику. Для того, щоб ефективно вирішувати проблеми, пов'язані з динамічними алгоритмами порівняння, потрібно використовувати тимчасові масштаби для двох слів по оптимальному відповідності [43].

Одним з напрямків, які пов'язані із залученням процедур, що стосуються автоматичного аналізу мови можна виділити завдання із здійсненням візуалізації результатів, що призводить до допомоги в навчанні вимові. Подібне навчання націлене як на окремі звуки, так і на цілі слова і фрази [44].

На даний момент досить поширеним способом, застосовуваним при вирішенні аналогічних завдань по аналізу і розпізнавання мови, можна вважати статистичний підхід. Здійснюється уявлення мовних одиниць в Гауссових моделях сигналів [45], приклад зображено на рисунку 1.11

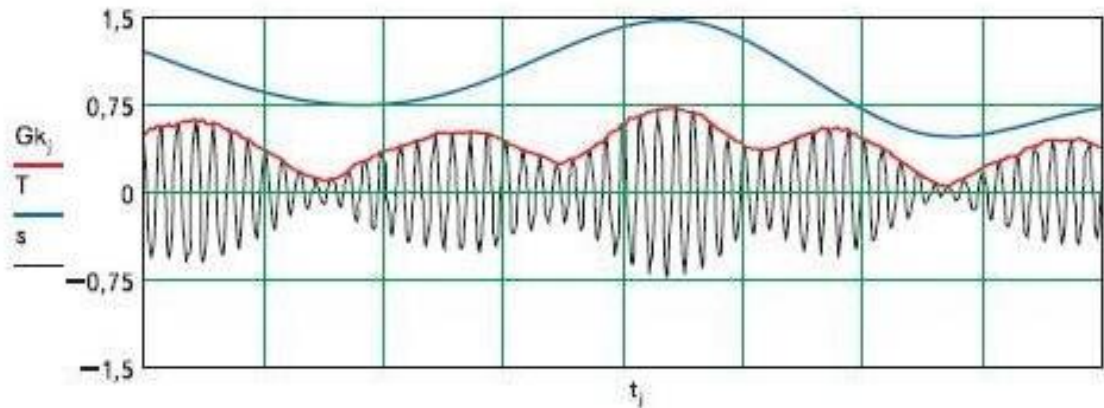


Рисунок 1.11 – приклад Гауссових моделей сигналів для мовних одиниць

Йде представлення кожного слова у вигляді одного або декількох еталонів по простору вимірювань і відбувається обчислення відстаней від еталонів по відношенню до невідомих реалізацій мовних сигналів. Для простого випадку, коли йде виникнення відхилень для реалізацій деяких слів від еталонів, спираючись на випадковий процес, який має нормальний розподіл, як оптимального можна вважати проведення обчислення відстаней в рамках евклідової метрики, яке реалізується, наприклад, як коефіцієнт кореляції [46]. Для того щоб проводити реалізацію адаптивних алгоритмів розпізнавання, потрібно використовувати мовні еталони, що дають можливості проводити відображення по повільних змін у вимові людей щодо часу. При реалізаціях важливо залучати зворотні зв'язки для рівнів точності по кожних спробах, пов'язаних з розпізнаванням [47].

Важливо розуміти, що при проведенні обробки текстових даних потрібно застосовувати комплексні інформаційні системи [48].

Консорціум Всесвітньої Павутини (W3C) розробив у 2004 році стандарти Граматик для розпізнавання і синтезу мови. Ці документи визначають основні характеристики мови. Згідно з цими стандартами можна класифікувати системи розпізнавання мови за такими ознаками:

- інтервал між окремими словами: якщо система розпізнає безперервну мову, користувач може виголошувати промову природно. У системах з дискретної промовою паузи між словами повинні становити не менше чверті секунди [49];

- залежність від диктора: системи розпізнавання, незалежні від диктора, дозволяють працювати без попередніх налаштувань, але дозволяють поліпшити надійність розпізнавання після навчання. Подібна незалежність досягається за рахунок зберігання звукових еталонів для всіх найбільш типових голосів носіїв мови [50];

- ступінь деталізації при завданні еталонів: різні алгоритми розпізнавання мови можуть використовувати для розпізнавання мови, як еталони слів, так і зразки мовних одиниць [51];

- розмір словника: розрізняють маленькі словники (близько 50 слів), що дозволяють передавати команди комп'ютера; середні (близько 1000 слів), достатні для визначення «активного» словника; великі (понад 10 000 слів) для диктування текстів [52].

У таблиці 1.1 представлені мовні моделі, які використовуються сьогодні в системах розпізнавання мови з необмеженими словниками.

Більшість сучасних систем розпізнавання мови використовують подібні алгоритми і методи [53]. Специфіка завдання і існуючі обчислювальні обмеження визначає різницю в типі диктування мови, розмірі словника.

Існують можливості підвищення швидкості при проведенні розпізнаванні мови, коли враховують такі критерії:

- розміри для тієї лексики, яку аналізують;
- час, необхідний для того, щоб адаптувати систему;
- ступінь спонтанності мови, яку потрібно розпізнати;
- час, необхідний для того, щоб здійснити розпізнавання [54].

Таблиця 1.1 – Моделі, які використовуються сьогодні в системах розпізнавання мови

Типи	Переваги	Недоліки
n-грами	Можливість побудови моделі по навчальному корпусу великого розміру. Висока швидкість роботи	Не враховується залежність чергового слова від розміру всього тексту
Моделі, засновані на деревах рішень	Теоретична здатність показати істотне поліпшення з n-програмних моделями	Потрібен великий обсяг інформації, що аналізується. Не відомо
Моделі, засновані на теорії формальних мов	Висока швидкість граматичного розбору	Ігнорування пропозицій, які не вкладаються в правила формальної мови
Адаптивні моделі	Можливість коригування моделі у міру роботи розпізнавача	Низька швидкість

Крім цього, існує ще ряд проблем:

- придушення стаціонарних і нестаціонарних перешкод;
- перехід до розпізнавання безперервної мови;
- облік контексту;
- пошук нових звукових параметрів;
- перенесення на інші мови;
- пошук нових алгоритмів відновлення послідовності вимовлених звуків у системі [55].

Дослідження проблем автоматичного розпізнавання мови є важливим фундаментальним напрямком. Ця проблема зараз стримує розвиток різних прикладних систем в телекомунікаціях, медицині, освіті і повсякденному житті при проектуванні [56].

Роблячи висновки цього підрозділу, можемо сказати, що вся сучасна техніка і різні сервіси використовують автоматизовані засоби управління і обробки інформації, системи розвиваються в напрямку підвищення комфортності

спілкування людини з комп'ютером, тому розробка ефективних засобів людино-машинного взаємодії є першочерговою науковою задачею [57].

#### **1.4 Варіантний аналіз і обґрунтування вибору програмних засобів для реалізації веб-системи**

У сучасному світі існує безліч мов програмування. Кожна мова має свої переваги і недоліки. Проаналізуємо найбільш популярні мови програмування, на яких є можливість створювати веб-додатки типу клієнт-сервер: C#, Java, JavaScript, PHP. Ми узагальним та порівняємо ці мови різними критеріями. Наприкінці ми зробимо свій висновок про те, які мови придатні і які не призначені для використання в деяких випадках. Також надано докази та наш аналіз того, чому деякі мови кращі за інші або мають переваги над інший за деяким критерієм [58].

Оскільки існує сьогодні сотні мов програмування, це неможливо і неефективно прикласти зусилля для аналізу кожної мови. Але ми можемо класифікувати деякі представницькі категорії мови та проводити глибокі дослідження за певними критеріями [59].

Проблема полягає в тому, щоб порівнювати та контрастувати 4 мови відповідно до 11 зазначених критеріїв.

Мета визначення придатності та придатності мов для кожного критерію, розрізнити їх плюси і мінуси, оцінку та вивчення пов'язаних з ними функцій на цих мовах.

В сучасних системах «Розумний дім» використовують веб-сервіси для віддаленого керування системою, через мережу Веб-сервіс – це програмна система, призначена для підтримки взаємодії між машиною та комп'ютером через мережу. Він має інтерфейс, описаний у форматі, здатному працювати з серверами (зокрема WSDL). Інші системи взаємодіють із веб-службою за допомогою SOAP-



повідомлень, які зазвичай передаються за допомогою HTTP із серіалізацією XML у поєднанні з іншими стандартами Інтернету. Отже, в основному веб-сервіси базуються на базовому наборі стандартів, що описують синтаксис і семантику спілкування. Тут XML забезпечує загальний синтаксис для представлення даних; SOAP (Simple Object Access Protocol) забезпечує семантику обміну даними; і WSDL (мова опису веб-служб) надає механізм для опису можливостей веб-служби. На відміну від традиційної клієнт-серверної системи, такі як веб-сторінка, веб-служби не надають користувачам графічний інтерфейс, а діляться бізнес-логікою та різною обробкою через інтерфейси в мережі [60].

#### **1.4.1 Мова C # та технологія ASP.NET Core**

Мова C # платформи .NET використовується для розробки веб-додатків у технології ASP.NET Core, де кожен файл ASPX може мати клас C#. Файл C#, який містить цей клас, який використовується для реалізації програмного забезпечення ASP.NET Core, називається кодом за файлом. Цей клас може містити ініціалізацію, обробників подій, допоміжні методи та коди тощо. Розроблені веб-додатки можуть бути розміщені на сервері Microsoft IIS, або крос-платформний веб-сервер Kestrel [61].

Веб-розробка з використанням C# виконується в IDE Visual Studio, яка забезпечує широку і багату підтримку для розробників, таких як веб-дизайнер, різні вбудовані елементи керування веб-серверами, рамки бібліотек тощо. Розробка з використанням C# є гнучкою, швидшою та зручною для програмістів.

.NET Framework забезпечує широку підтримку сумісності через веб-служби. У C#, використовуючи .NET Framework, Visual Studio та ASP.NET, створити веб-службу так само просто, як просто створити консольний проект. Створення Web-сервісу простіше в C# за допомогою .NET Framework та IDE Visual Studio. Компіляція та розгортання веб-сервісу легка і швидка [63].

#### **1.4.2 Платформа Java EE та технологія Java Persistence API**

Java має здатність створювати високоякісні веб-додатки. Java має API для створення цих програм. Компоненти використовуються як контролер, та можуть бути використані як модель, а JSP використовується як шаблон для перегляду.

Java-компоненти Enterprise можна використовувати як модель, яка може бути розташована в розподіленому середовищі.

Платформа Java EE підтримує безпеку, автентифікацію, авторизацію, транзакцію. Управління з'єднанням з базою даних за допомогою JNDI API забезпечує гнучкість. Java API для обробки XML (JAXP), що є частиною платформи Java SE, підтримує обробку XML-документів за допомогою моделі об'єкта документів (DOM). JAXP дозволяє програмам аналізувати та перетворювати XML-документи незалежно від конкретної реалізації XML-обробки [64].

Java Persistence API - це технологія Java, основана на стандартах на основі завзятості. Вона створена для подолання розриву між об'єктно-орієнтованою моделлю і реляційною базою даних [65].

### **1.4.3 PHP-скрипти**

PHP чудово підходить для веб-програм. PHP-скрипти виконуються на сервері. PHP підтримує безліч баз даних (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC та ін.). Одна з основних причин, чому PHP настільки популярний – це програмне забезпечення з відкритим кодом, яке можна безкоштовно завантажувати та використовувати. PHP в поєднанні з MySQL є крос-платформною (ви можете розробляти як в Windows та і на платформі UNIX). PHP працює на різних платформах (Windows, Linux, UNIX та ін.). PHP сумісний практично з усіма веб-серверами, що використовуються сьогодні. Однак, з появою Web 3.0, де програмування використовує в основному Ajax, що більше стосується до JavaScript, XML та CSS, PHP почав втрачати перші позиції в рейтингу найпопулярніших мов для серверного програмування [66].

Веб-сервіс складається з сервера, який обслуговує запити до веб-сервісу, а клієнт - за допомогою методів веб-сервісу. Бібліотека класів PHP надає розширення SOAP для розробки серверів і клієнтів SOAP та розширення XML-RPC для створення серверів і клієнтів XML-RPC.

Важливою перевагою веб-сервісів є повсюдність між платформами та мовами. Сценарій PHP, що працює на Linux, може спілкуватися з сервером IIS на Windows, використовуючи ASP без проблем зв'язку [67].

#### **1.4.4 JavaScript та об'єкт XMLHttpRequest**

JavaScript - це скриптова мова, яка працює як частина веб-браузера. В веб-програмах, в основному, він використовується на стороні клієнта, щоб забезпечити перевірку даних та взаємодію з моделлю об'єктів документів, вставляючи його в теги HTML. Такий підхід для роботи на стороні клієнта називають ще JavaScript (CSJA).

Тепер Javascript (SSJA) почав використовуватися і на стороні сервера. Перша реалізація SSJS - це програма LiveWire Netscape, яка була включена до Enterprise Server 2.0 ще в 1996 році. З тих пір ряд інших компаній застосував його, запропонувавши альтернативу звичайним технологіям на стороні сервера, такими як ASP Microsoft та інші. Ці системи Javascript підтримують доступу до бази даних, файлової системи, відправлення електронної пошти тощо.

Таким чином, використовуючи Ajax API, веб-додаток може спілкуватися асинхронно з веб-сервером в інтерактивному режимі. Це повідомлення дозволяє отримувати дані з веб-сервера за допомогою об'єкта XMLHttpRequest.

JavaScript дозволяє викликати веб-служби, використовуючи Ajax (об'єкт XMLHttpRequest) для отримання настроєних даних. Yahoo! і веб-сервіси Amazon є популярним прикладом для створення клієнтських веб-сервісів на основі XMLHttpRequest в JavaScript [68].

Отже провівши огляд існуючих мов програмування для веб-додатків зведемо данні до таблиці 1.2

Таблиця 1.2 – Особливості існуючих мов програмування для веб-додатків

Особливості	C#/ASP.NET Core	Java EE	PHP-скрипти	JavaScript
Динамічні веб-сторінки	Так, як код на веб-сторінках ASP.NET Core	JSP	Підтримка, код за веб- сторінкою php	Використання серверного JavaScript(SSJS)
Веб-сервер	IIS на ОС Windows, Kestrel (Apache, Nginx)	Tomcat, JBoss, GlassFish	Apache, Microsoft IIS	IIS, Apache
Web Framework/ бібліотеки	Використання .NET Framework для ОС Windows і ASP.NET Core Linux та Unix.	J2EE framework	Багато (наприклад, Yii2, Symfony тощо)	JAXER
Управління сесією	Так, управлінням Станом ASP.NET Core	Так, вони представлені об'єктом HttpSession.	Так, використовуючи \$_SESSION змінну	Так, використовуючи Jaxer.session
Безпека	Так (Архітектура безпеки ASP.NET Core)	Так	Не настільки безпечно	Так
Model-View Controller	ASP.NET MVC 5/ Core MVC	J2EE /Spring/ Struts	Symfony/Mojavi /CakePHP	Так, MVC за допомогою Jaxer
Взаємодія бази даних	Так, використовуючи бібліотеку ADO.NET, або Entity Framework	Так (JDBC)	Mysql	Так, Jaxer.DB
Підтримка рівня безпеки Secure Sockets Layer(SSL)	Так, сертифікат SSL може бути створений та використаний для сервера IIS.	Так, JavaSSL	Так (використовуйте відкриту SSL-функцію)	Так
Веб-сервіси(SOAP, WSDL, UDDI)	Так.використовуючи веб-сервери ASPT.NET для Windows і Web Services проекту Core на інших ОС (Linux / Unix, Mac).	Так. Тобто за допомогою Apache XML-RPC Для бібліотеки Java або Spring Web Services, JAX-WS).	Так (XML-RPC для PHP)	Так, можна використовувати клієнт веб-сервісу на основі AJAX / XMLHttpRequest.

Продовження таблиці 1.2

Особливості	C#/ASP.NET Core	Java EE	PHP-скрипти	JavaScript
Безпека веб-служб	Так, шифрування та підпис HTTPS за допомогою SSL стандартів	Так. HTTPS, підтримує "стандарти, основані на XML", такі як WS-Policy, WSSecurity та WSTransfer стандарти)	Часткова підтримка (обмежена взаємодією SOAP з WSDL)	Так, Аїах можна використовувати з будь-якою динамічною мовою веб-програмування, що підтримує протокол HTTPS (SSL).
Композиція веб-служб	Використання веб-сервісів ASP.NET Core. Також можливо, зв'язати два або більше Web Services, які обмінюються даними між собою.	Композиція J2EE WS.	Композиція J2EE WS.	Ні, JavaScript не підтримує композицію веб-служб. Ми можемо використовувати тільки веб-сервіси, використовуючи JavaScript.

Отже розібравши особливості існуючих мов програмування для веб-додатків, було прийнято рішення використовувати C#/ASP.NET Core, через такі переваги, як:

- крос-платформна технологія;
- безпека даних;
- підтримка багатьох веб-сервісів;
- взаємодія з базою даних, може виконуватися як через складний, та оптимізований ADO.NET; так і через простий та гнучкий Entity Framework;
- підтримка зв'язування веб-сервісів, яка просто та надійно реалізована [68].

## 1.5 Вибір напрямку та задач дослідження

Об'єктом управління буде будинок, який зображено на рисунку 1.12. Після дослідження проблем розпізнавання голосу, було прийнято рішення по вдосконаленню існуючих алгоритмів, для підвищення швидкості розпізнавання. Веб сервіс буде написано на ASP.NET Core. Для керування будинком буде використано мікроконтролер Arduino на базі Atmel 328.

Основними характеристиками системи будуть:

- зручність використання;
- швидкість передавання розпізнаних команд до контролера;
- швидкість розпізнавання голосових команд;
- безпека передачі даних між клієнтом та сервером

Головними завдання контролера повинно бути:

- управління освітлення;
- управління кліматом;
- спостереження за безпекою;
- сприйняття голосових команд від веб-сервісу додатка;
- передавання інформації про стан будинка до веб-сервісу.

Основними параметрами (таблиця 1.3), які будуть контролюватися в розумному будинку будуть температура, вологість та рівень освітлення. Для параметру температури існують допустимі норми верхньої та нижньої границі, регулювання буде здійснюватися за допомоги ввімкнення опалення та кондиціонеру. Якщо температура перевищує допустимі норми то вмикається опалення, до того часу поки температура не стане комфортною, або користувач вимкне систему, якщо навпаки температура надто висока, система включить кондиціонер. Для контролювання вологості в кімнатах будуть встановленні спеціальній зволожувач та очищувач повітря. Контроль рівня освітлення буде реалізований за допомоги світлових резисторів. Також будуть присутні датчики присутності [69].

Таблиця 1.3 – Керовані параметри

Керовані параметри	Скорочення	Одиниця виміру	Норма	Допустимі відхилення
температура	<i>T<sub>кім</sub></i>	0 <sub>с</sub>	25	±2
вологість	<i>V<sub>кім</sub></i>	%	55%	±15%

## 2 ПРОЕКТУВАННЯ ГОЛОСОВОЇ СИСТЕМИ УПРАВЛІННЯ «РОЗУМНИЙ ДІМ»

### 2.1 Аналіз функцій комп'ютеризованої системи управління «Розумний дім»

В даній роботі потрібно розробити систему управління «розумний дім», яка матиме класичні для подібних систем 3 рівні. Перший рівень буде в якості клієнта, та містити голосове розпізнавання, та передавати розпізнані команди на другий рівень. Другий рівень буде в якості веб-сервісу, та буде взаємодіяти з першим рівнем та третім рівнем. Цей рівень створюється для віддаленого користування, або так званої технологія, як інтернет речей.

Веб-сервіси, які виступають на другому рівні мають ряд переваг, одна з найчастіше застосованих в сучасних системах – це те що на перший та третій рівень можуть змінитися, наприклад зміняться обладнання на третьому рівні (вийшов новий більш потужний мікроконтролер), або операційна система на першому (власник змінив свій смартфон з Android OS на Mac OS). Превагою ще можливо вважати, те що весь написаний код, скритий від користувачів системи, це надає більше безпеки для системи. Третій рівень виступає у ролі контролера, який буде виконувати вказівки надіслані з веб-сервісу (тобто другого рівня) [70].

В загальному цілому система легко масштабується, рівні можуть змінюватися та доповнюватися. Інтеграція бази даних, якщо потрібна власниками

системи, відбувається на другому рівні. Вона також буде повністю захищена від споживачів системи.

Безпека системи, полягає також в використанні Token-ів, або як ще кажуть «ticket», або білет. Для того щоб отримати Token, потрібно пройти авторизацію системі за допомоги логіна та паролю. Авторизація відбувається на веб-сервісі, а логін та пароль вводиться у клієнті. Якщо авторизація вдала, то на клієт приходить JSON файл, в якому міститься Token, якщо авторизація не вдала, у JSON файл передається повідомлення, в якому вказано причину не отримання Token-у. Далі усі функції системи які відправляються з клієнта, мають містити унікальний Token, який був отриманий при авторизації. Завдяки використанню Token-ів, безпека в системі стає високою і не можливою для доступу третіх осіб.

В загальному система повинна виконувати такі функції:

- клієнт:

- 1) розпізнавання голосових команд;
- 2) відображення стану системи;
- 3) під'єднання до веб-сервісу;

- веб-сервіс:

- 1) проведення авторизації клієнтів;
- 2) обмін даними між клієнтом та мікроконтролером;

- мікроконтролер:

- 1) виконання команд отриманих від веб-сервісу;
- 2) відправка поточного стану на веб-сервіс, якщо надійшов запит;
- 3) вимірювання та регулювання температури та вологості [71];

Акторами у даній системі є користувач, клієнт, веб-сервіс, мікроконтролер.

Користувач виконує наступні функції: говорить команди та спостерігає за поточним станом.

Клієнт виконує розпізнавання голосових команд, надає можливість спостерігати за поточним станом, взаємодіє з веб-сервісом.

Веб-сервіс в свою чергу виконує обмін даними, перевіряє Token, генерує унікальний Token.



Мікроконтролер виконує команди отримані з веб-сервісу, відправляє данні на веб-сервіс, вимірює температуру, вологість та регулює температуру та вологість. UML-діаграма прецедентів подана на рисунку 2.1

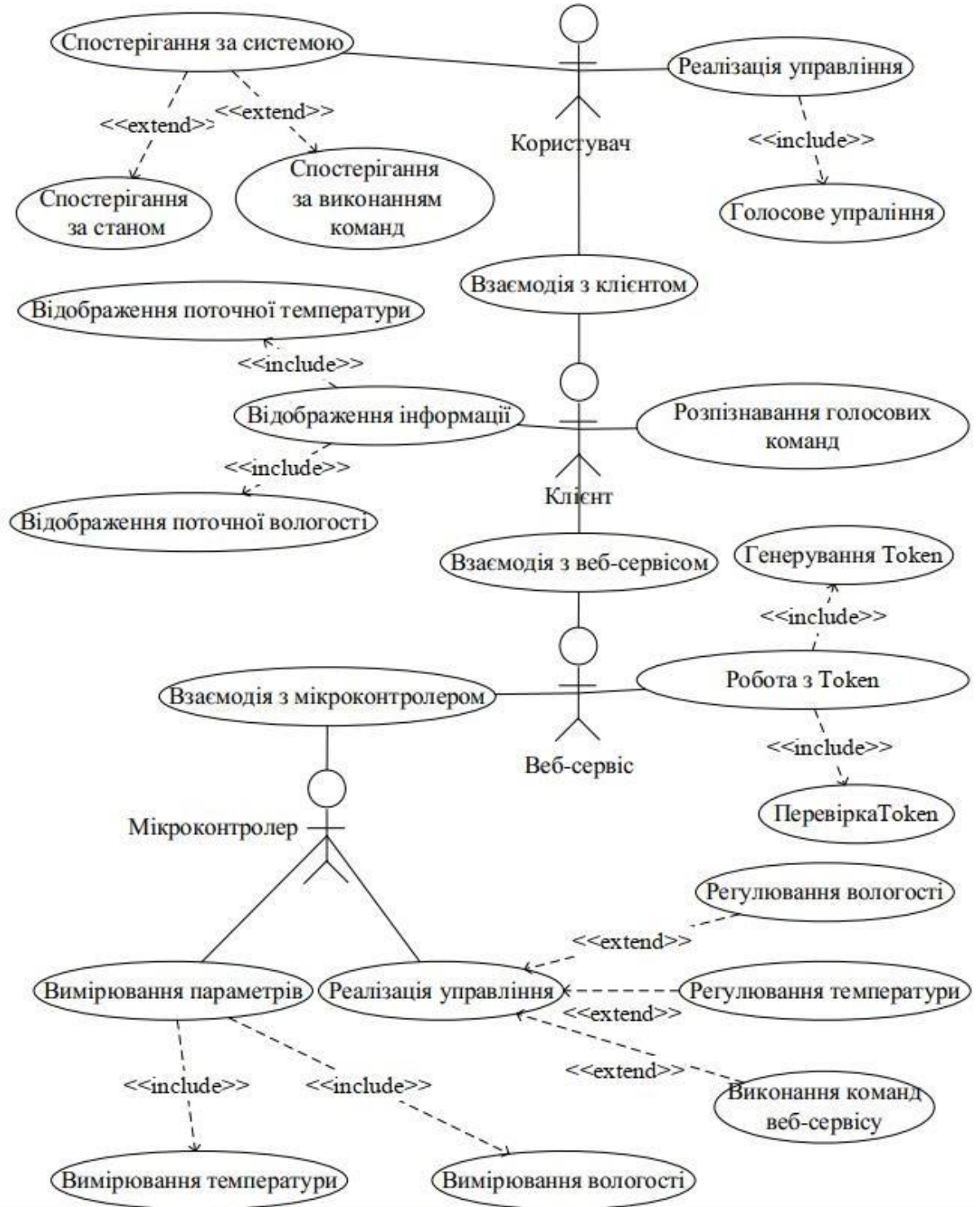


Рисунок 2.1 – UML-діаграма прецедентів

## **2.2 Розробка структурної схеми КСУ розумним домом**

Розроблена система повинна вимірювати значення температури, якщо температура перевищує оптимальну, то ввімкнеться реле, яке в свою чергу включить систему охолодження, якщо температура опуститься за нижню допустиму границю, тоді ввімкнеться система обігріву. Також система повинна виконувати голосові команди. Це можна здійснити за допомогою клієнта, веб-сервісу, мікроконтролера. Останній з названих буде вмикати та вимикати спеціалізовані реле.

Для нормальної роботи системи данні параметри повинні бути в межах нормованих значень. Для приведення поточних значень параметрів до нормованого проводиться розрахунок управляючого впливу [72].

Реалізація розрахованого впливу здійснюється за допомогою мікроконтролера. Регулювання здійснюється неперервно, або за допомоги голосового управління. Розроблена структурна схема даної системи зображена на рисунку 2.2.

## **2.3 Розробка схеми автоматизації КСУ розумним будинком**

Основними факторами при роботі апарату для розумного будинку є:

- температура у кімнатах;
- вологість у кімнатах.

Вище наведені параметри вимірюються за допомогою датчиків, результати яких передаються на мікроконтролер. Результати виміряних параметрів, передаються на регулюючі пристрої, які вираховують регульований вплив та передає на виконавчі механізми, в даному випадку це спеціалізовані реле.

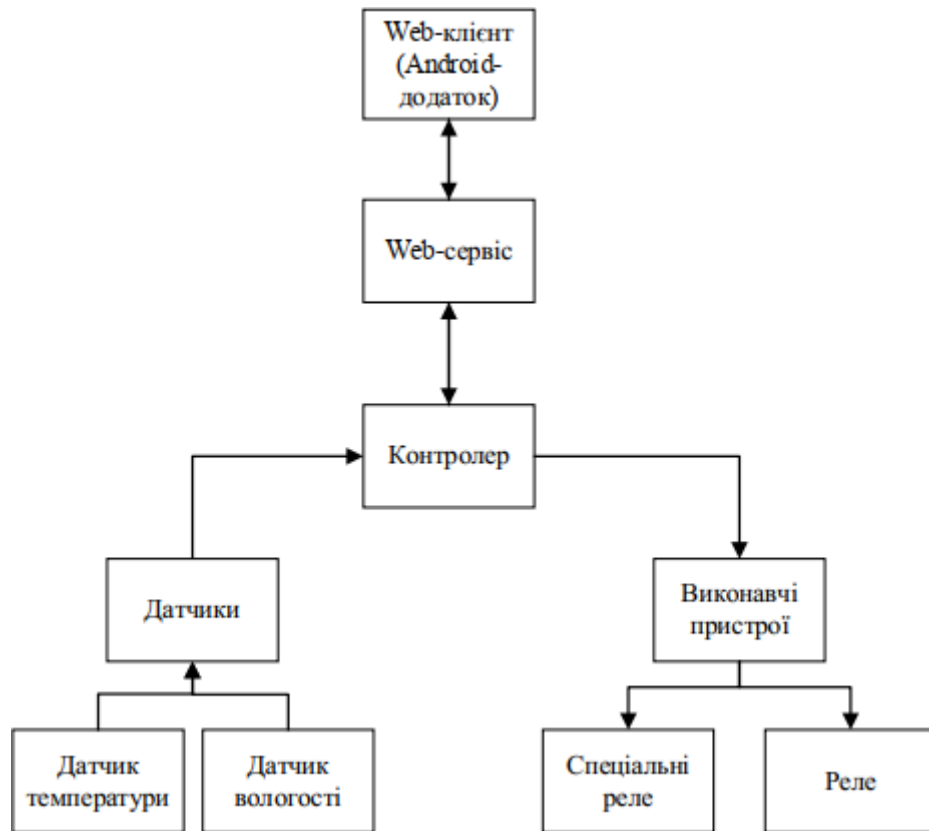


Рисунок 2.2 – Структурна схема системи

Також поточні данні передаються не веб-сервіс, який розташований далеко від будівлі споживача. Зв'язок між веб-сервісом та мікроконтролером відбувається за допомоги мережі. Веб-сервіс в свою чергу взаємодіє з клієнтом, також через мережу, який передає розпізнані голосові команди. На рисунку 2.3 подана функціональна схема даної системи.

#### 2.4 Дослідження та удосконалення алгоритму розпізнавання голосу

Більшість методів розпізнавання звичайно працюють зліва направо, розміщуючи лише одну фонему за один раз в кінці фактичного префіксу слова. Ці

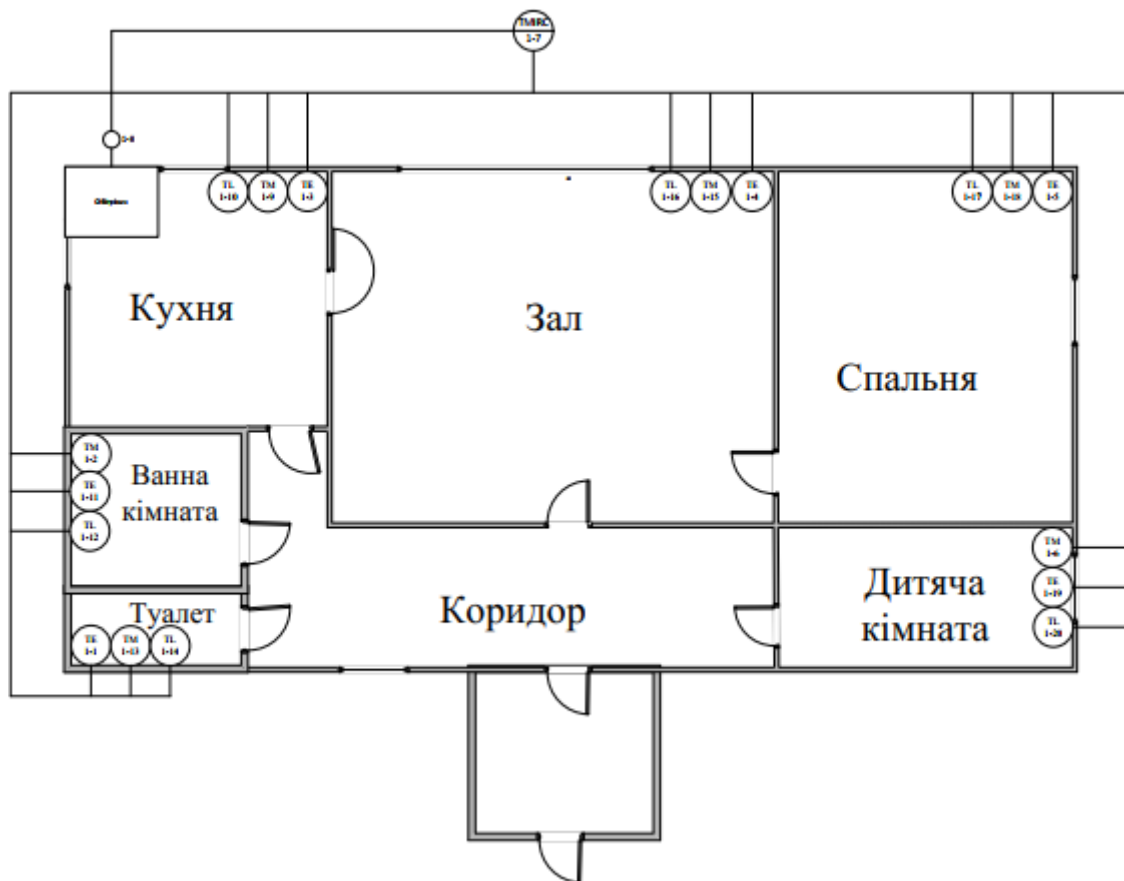


Рисунок 2.3 – Функціональна схема КСУ розумним домом

префікси - це так звані гіпотези, які є об'єктними парами у формі  $(o_1 \dots o_j, [A_1, \dots, A_j])$ . Зшивання іншої (дозволеної) фонем та нового закінчення часу до гіпотези називають його продовженням. Два основних алгоритмів пошуку, які ми будемо застосовувати, - це алгоритм багатостадійного декодування та метод пошуку в пучку Вітербі; їх можна розглядати окремо. Обидва можуть бути реалізовані, призначаючи структури даних (стеки) у будь-яке можливе місце, де може бути зв'язок між фонем; ці стеки (не плутати зі стеками LIFO чи FIFO) просто зберігають гіпотези на підставі їх ймовірності. Стовп обмеженого розміру містить лише найбільш вірогідні, і відкидає найгірші, що перевищують його потужність  $n$ . Алгоритм множинного декодування використовує стеки обмеженого розміру ( $n$  є єдиним параметром методу), тоді як алгоритм пошуку в пучку Вітербі зберігає лише найкращу гіпотезу в стеку, а ті, що є близькими до нього, що контролюється параметром ширина промені. В іншому випадку два алгоритми пошуку можна

розглядати як такі, що працюють однаково. Спочатку початкова гіпотеза ставиться в стек, присвячений початку мовного сигналу. Потім ми розглядаємо стек у порядку збільшення часу: для кожного кроку ми отримуємо всі гіпотези від поточного стека, розповсюджуємо їх всіма можливими способами та покладаємо сформовані гіпотези на стек відповідно до їх нових закінчень. Результатом цього методу є найбільш вірогідна гіпотеза останнього стека з послідовністю фонем, що належить слову зі словника [73].

Як голосовий сигнал з великою випадковістю, навіть якщо одна і та ж особа різний час мовить одним голосом, вона не може мати точно таку ж тривалість часу, тому важливо знешкодити динамічний час. Для вирішення цієї проблеми найсучаснішою технологією є техніка динамічного деформування часу (динамічне деформування часу, DTW). DTW - це нелінійна реформована технологія, яка поєднується з розрахунком деформування часу та відстані. Розпізнавання мови також успішне в алгоритмі відповідності. Динамічне деформування часу перетворює комплексну глобальну задачу оптимізації на багато локальних задач оптимізації

рисунк 2.4, використовуючи методи динамічного програмування (динамічне програмування, ДП), прийняття рішень поступово. Припустимо, що функція векторної послідовності еталонних шаблонів виглядає наступним чином

$X = \{x_1, x_2, \dots, x_i\}$ , та векторна послідовність –  $Y = \{y_1, y_2, \dots, y_j\}, i \neq j$ .



Рисунк 2.4 – Використання динамічних доменів в системі

Алгоритм DTW - знайти оптимальну функцію деформування часу, яка мінімізує загальну суму кумулятивних спотворень та нелінійно відображає часову

шкалу невідомого еталонного шаблону до часової шкали і еталонного шаблону [74].

Експериментальна база даних виділяє 10 чоловічих та 10 жіночих голосів, які записуються за допомогою запису програмного забезпечення системи ПК Windows, на якій встановлено тестову систему розпізнавання голосових команд та систему одного зі відомих виробників. Результати швидкості відомого виробника – будемо вважати як 100%. Існує шум низької інтенсивності в середовищі запису. Експеримент був поділений на два процеси, навчання та визнання. Навчальний процес, тобто експеримент, динамік записує слово зі швидкістю повільніше, ніж нормальна швидкість мовлення, і збережить їх для оригінального шаблону, витягніть функцію WFCC, а потім архівуйте її. У процесі розпізнавання ми записали слова, використовуючи нормальну швидкість мовлення, одержували одночасно функцію WFCC, а потім підбирали їх за допомогою алгоритму DTW. Також рівень розпізнавання в порівнянні з MFCC, наступна таблиця статистики: Ми можемо отримати деяке рішення, засноване на таблицях: WFCC має покращений рівень розпізнавання в ізолюваних словах і ізолюваних слова в порівнянні з MFCC [75].

Отже зведемо данні в таблицю 2.1.

Таблиця 2.1 – Зведені дані експеременту

Метод	Оптимальні параметри	Правильність	Швидкість
Basic method		98.72%	100%
SMR для (MFCC)	$i = 10, j = 17$	99.45%	97.87%
SMR для (WFCC)	$i = 8, j = 28$	95.29%	78.29%
DTW для (MFCC)	$i = 15, j = 34$	93.22%	137.02%
DTW для (WFCC)	$i = 11, j = 24$	96.37%	155.57%

Як ми бачимо, с таблиці, то в ході експеременту, кращий по швидкості, в півтори рази швидше, ніж швидкість одного з відомих виробників, вийшов DTW

виростанням функції (WFCC), але в нього є недолік, правильність розпізнавання, слабкіша. На рисунку 2.5 зображено детальний алгоритм роботи цього методу. Саме цей алгоритм буде використовуватися на клієнтах нашої системи [76].

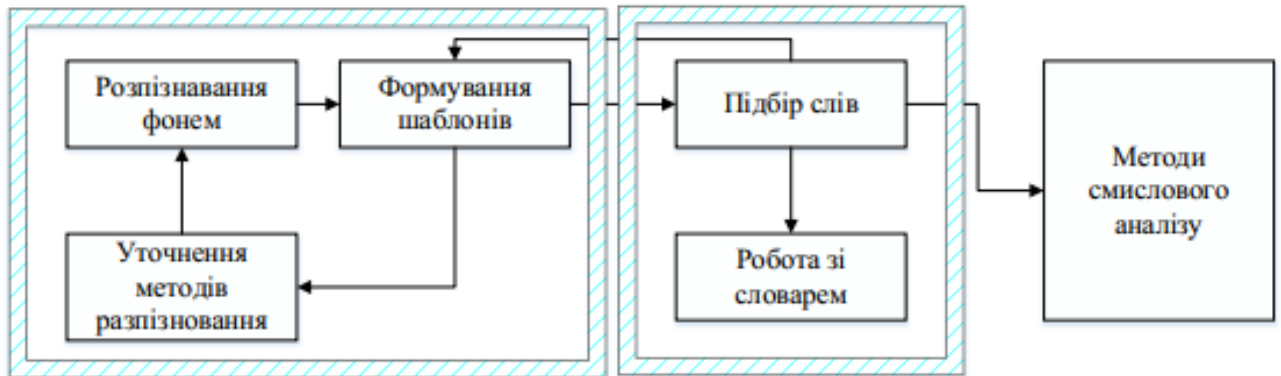


Рисунок 2.5 – Схема алгоритму

## 2.5 Розробка UML-діаграм функціонування КСУ розумним домом

UML-діаграма прецедентів – відображає відношення між акторами та прецедентами.

UML-діаграма діяльності – описує складові частини деякої діяльності.

У даному технологічному процесі керуючими впливами є:

- температура в кімнатах;
- вологість в кімнатах;

На вході в систему відбувається вимірювання перерахованих вище впливів. Якщо виміряне значення потрапляє в нормований проміжок, система перевіряє чи показник не досяг нормованого значення. Якщо виміряне значення не потрапляє в проміжок – розраховується регульований вплив, а потім виконується. Також користувач має змогу через голосові команди та клієнт впливати на виконавчі механізми, а саме реле [77].

Так як в нас система має три окремих рівня, то UML-діаграма діяльності, слід рознести на три окремі діаграми (рисунок 2.6–2.8), так як в підтримці програмного забезпечення функціонал має змогу змінюватися, та доповнюватися. UML-діаграма послідовності – це діаграма, на якій зображена впорядкована в часі взаємодія. Саме на цій діаграмі, описаний цикл взаємодіє системи, та послідовність виконання функцій трьох рівнів. На рисунку 2.9 подана UML-діаграма послідовності.

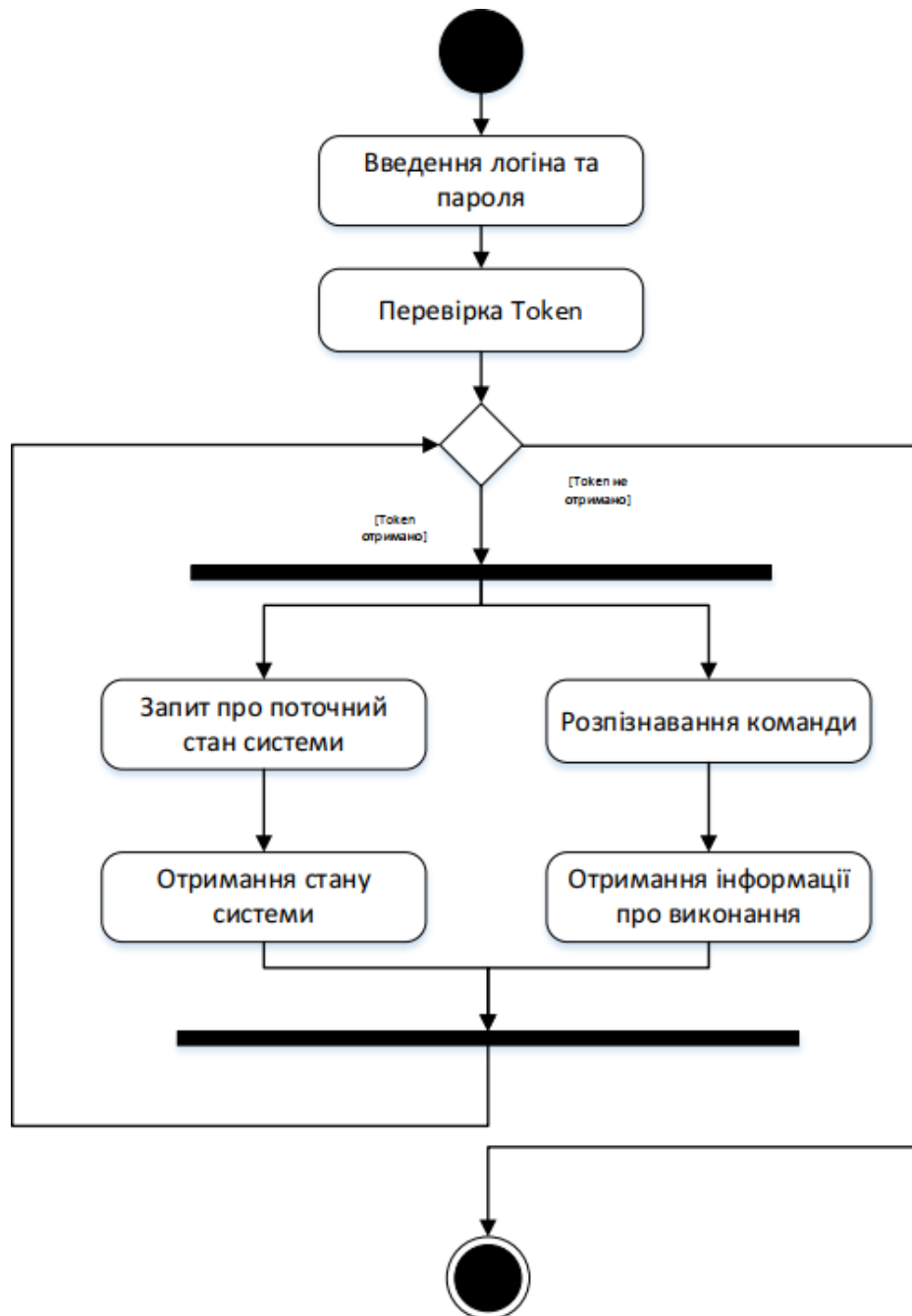


Рисунок 2.6 - UML-діаграма діяльності клієнта



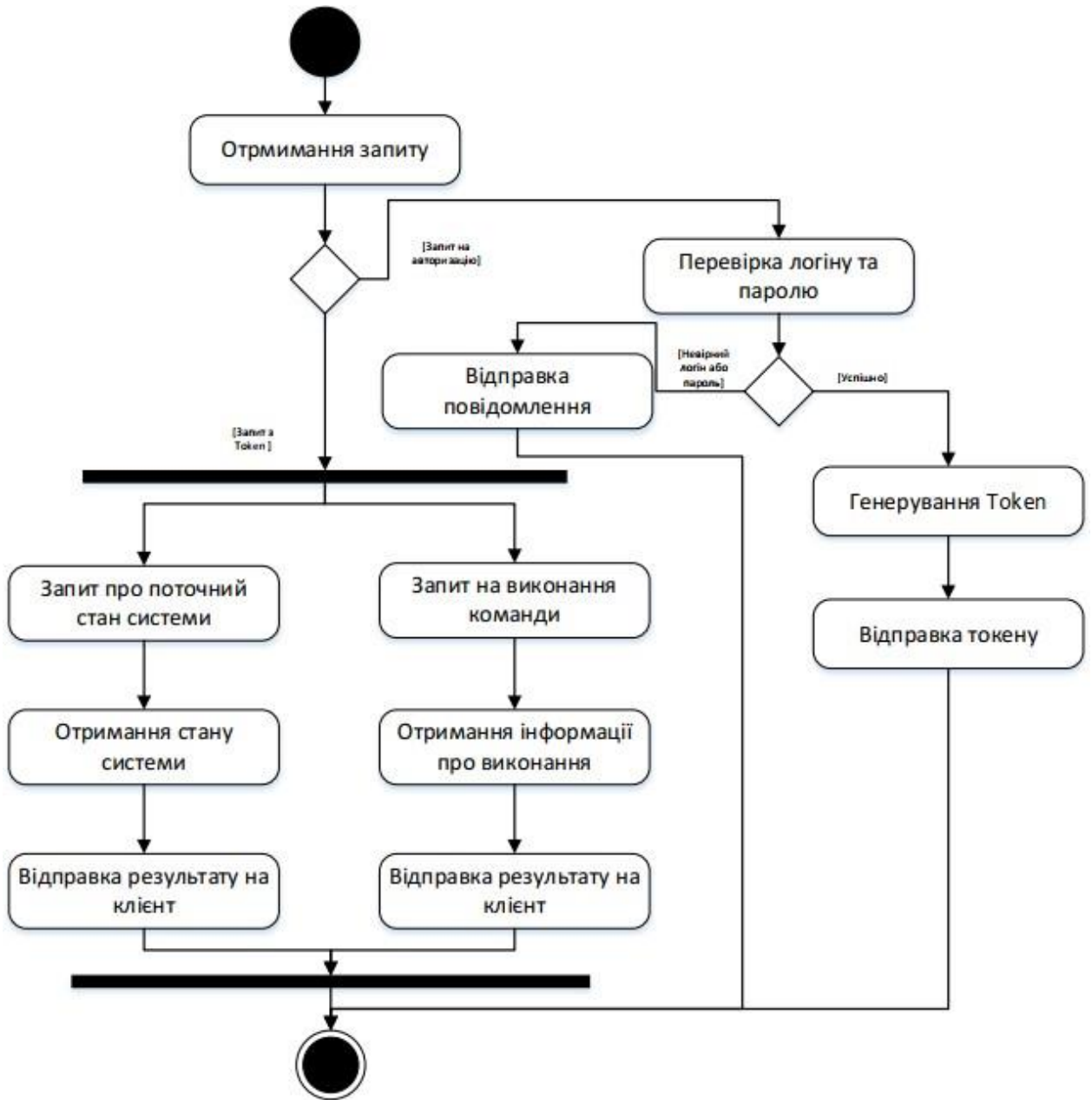


Рисунок 2.7 – UML-діаграма діяльності веб-сервісу

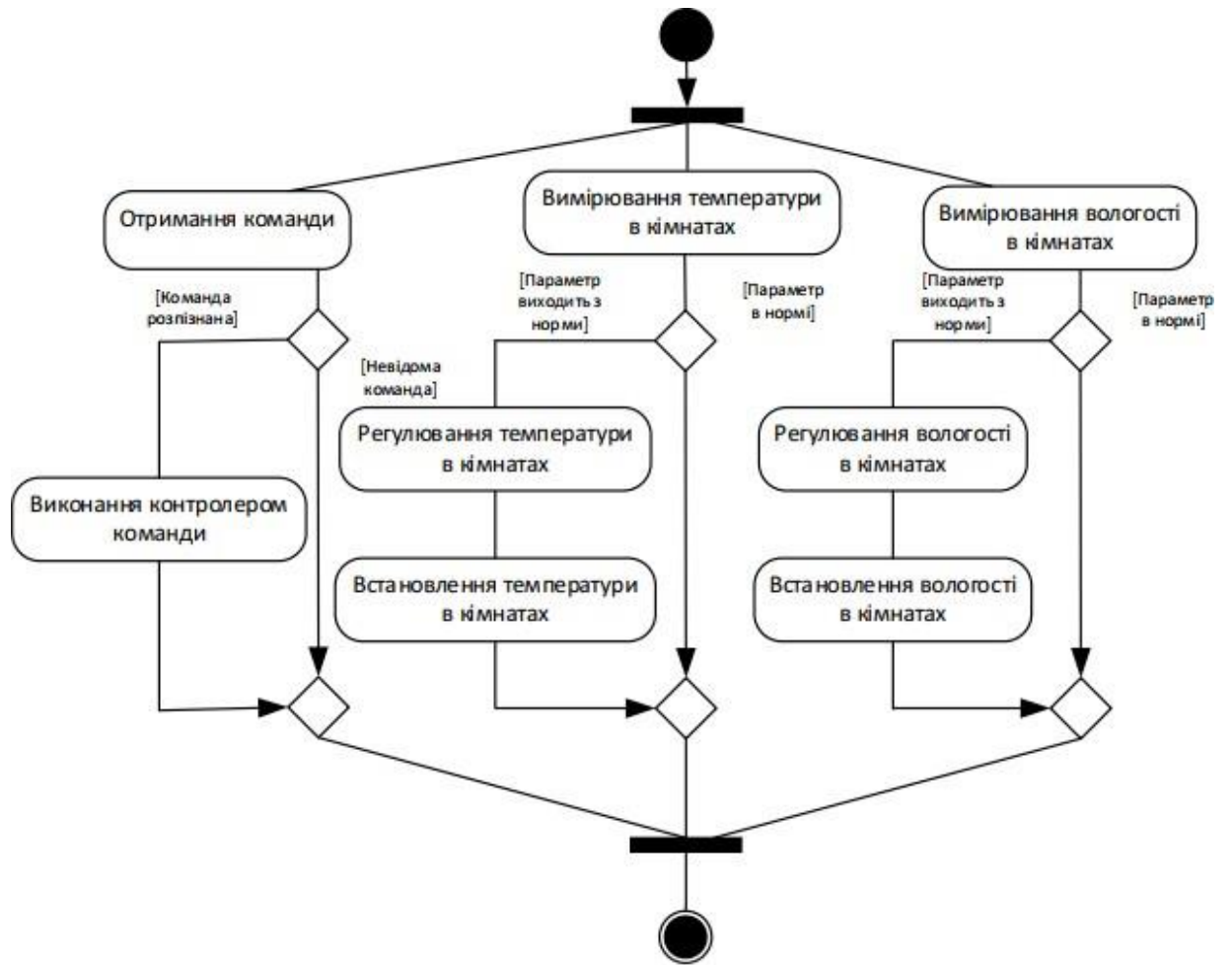


Рисунок 2.8 – UML-діаграма діяльності мікроконтролера

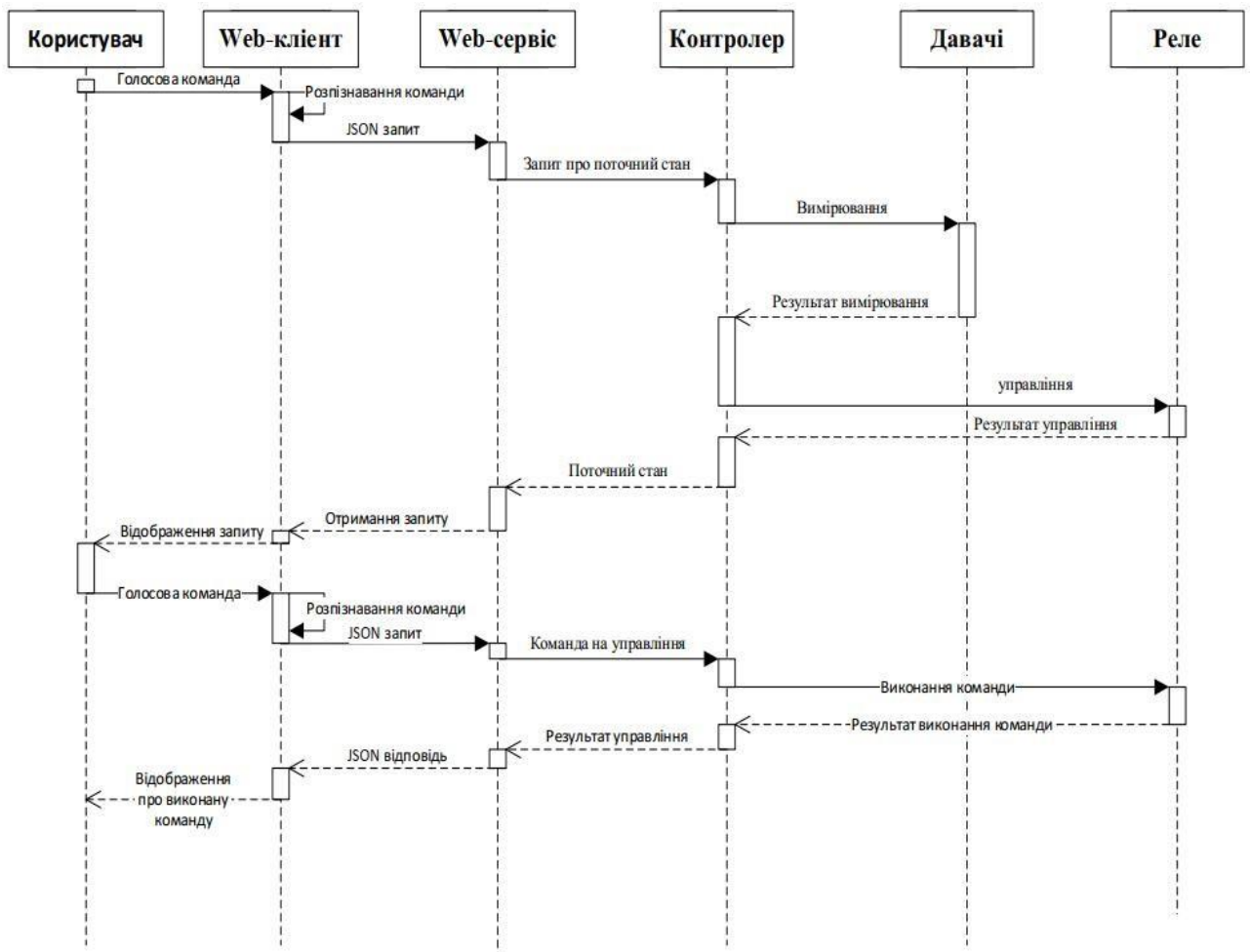


Рисунок 2.9 – UML-діаграма послідовності

## 3 РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ «РОЗУМНИЙ ДІМ»

### 3.1 Обґрунтування технічних засобів КСУ розумним домом

При виборі приладів контролю і регулювання керуються такими положеннями:

- прилади повинні забезпечувати необхідну точність вимірювання, бути досить чутливими та надійними в роботі;
- покази приладів повинні мати наочну шкалу і покажчик. Самописні прилади повинні регулювати свідчення у вигляді чіткої, добре помітною кривої;

- місцеві прилади повинні мати місця розташування, легко доступні для спостереження за показаннями;
- похибка не повинна виходити за доступні межі при зміні зовнішніх умов, температури і тиску навколишнього середовища;
- до вимірювальних і регулюючих приладів повинні пред'являтися вимоги по вибухо- і пожежобезпеці [78].

При виборі приладів контролю і регулювання повинні враховуватися властивості об'єктів регулювання та регуляторів, щоб системи регулювання були стійкими, і процес регулювання протікав якісно, без великих відхилень регульованої величини від заданого значення.

У даному технологічному процесі, для контролю, потрібні такі датчики:

- датчики температури для вимірювання температури;
- датчики вологості та вологості у кімнатах;
- спеціальні реле.

В якості датчика температури та вологості було обрано DHT22, тому що цей давач здійснює виміри температури та вологості водночас та має наступні характеристики:

- діапазон вимірювання вологості: 0-100%;
- діапазон виміру температури:  $-40 \sim 80 \text{ }^{\circ}\text{C}$  - для вимірювання температури кімнати цього діапазону достатньо;
- точність вимірювання вологості:  $\pm 2\% \text{ RH}$ ;
- точність вимірювання температури:  $\pm 0.5 \text{ }^{\circ}\text{C}$ ;
- низьке енергоспоживання;
- не вимагає обв'язки;
- довгий час життя;
- цифровий;
- низька ціна – 2\$ за один давач [79].

Завдяки цим характеристикам і було обрано цей давач який зображено на рисунку 3.1



Рисунок 3.1 – Давач температури та вологості DHT22

Arduino Ethernet Shield 2 з'єднує Arduino з мережою. Просто підключається цей модуль до плати Arduino. До мережі підключається за допомогою кабелю RJ45. Як завжди в Arduino, кожен елемент платформи - апаратне забезпечення, програмне забезпечення має документацію та є у вільному доступі з відкритим вихідним кодом. Це означає, що ви можете точно дізнатися, як це зробити, і використовувати його дизайн як відправну точку для власних схем.

В якості Arduino Ethernet Shield було обрано Контролер Ethernet: W5500 (рис.3.2) тому що, він має 32к буфер, якого достатньо для системи «Розумний дім», та має наступні характеристики:

- робоча напруга 5 В (постачається з плати Arduino);
- з внутрішнім 32К буфером;
- швидкість підключення: 10/100Мб;
- з'єднання з Arduino на порту SPI [80].

В якості спеціальних реле обираємо реле фірми SONGLE SRD-05VDC. Так як, ці прилади показали високий рівень надійності, в подібних системах. Реле потребує 5V, та може управляти приладами, які потребують високу напругу до 250V та силу току в 10A.



Рисунок 3.2 - Контролер Ethernet: W5500

На рисунку 3.3 відображено схему реле, а на рисунку 3.4 вигляд. Між контактами A1 і A2 встановлений металевий сердечник, при протіканні струму по якому до нього притягується рухливий яркір (2). Контакти же 1 і 3 нерухомі. Варто зазначити що яркір під пружин і поки ми не пропустимо струм через сердечник, яркір буде утримується притиснутим до контакту 3. При подачі струму, як уже говорилося, сердечник перетворюється в електромагніт і притягується до контакту 1. При знеструмленні пружина знову повертає яркір до контакту 3 [81].

Існує безліч мікроконтролерів і платформ для здійснення «physical computing». Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard і багато інших, які пропонують схожу функціональність. Всі ці пристрої об'єднують розрізнену інформацію про програмування і укладають її в просту у використанні збірку. Arduino, у свою чергу, теж спрощує процес роботи з мікроконтролерами, однак має ряд переваг перед іншими пристроями для викладачів, студентів та любителів:

- низька вартість - плати Arduino відносно дешеві в порівнянні з іншими

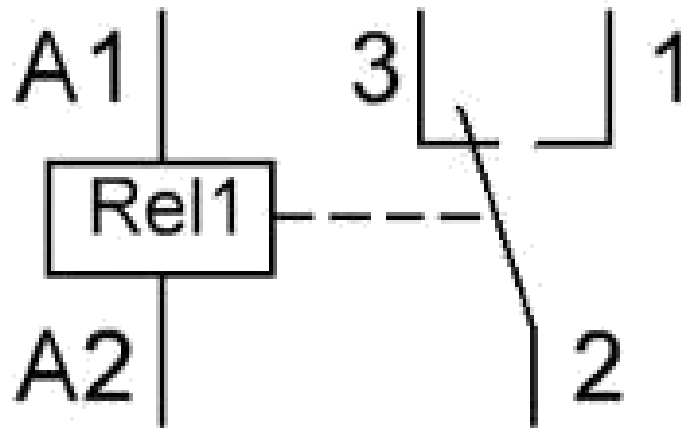


Рисунок 3.3 – Схема реле

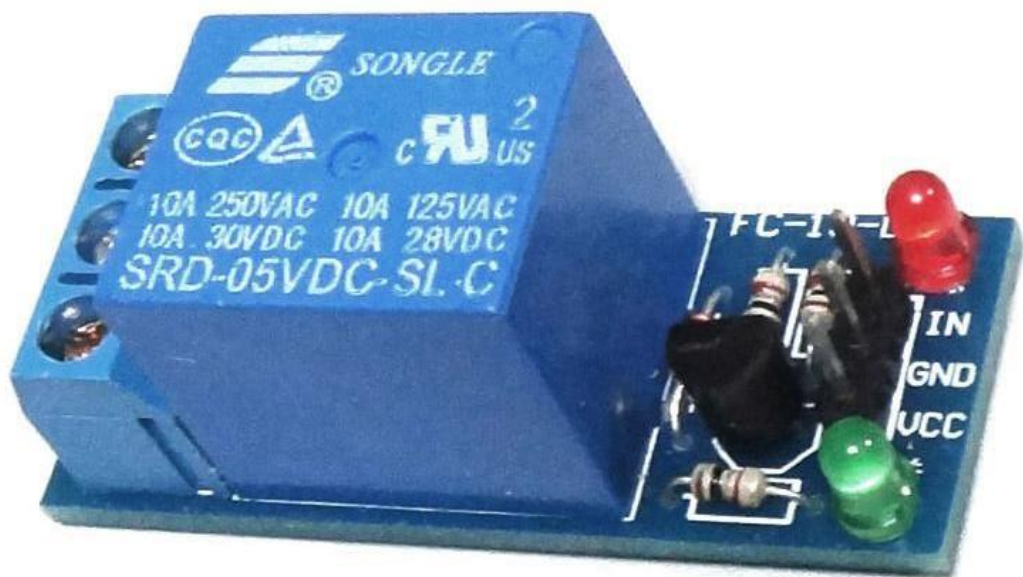


Рисунок 3.4 – Реле фірми SONGLE SRD-05VDC

платформами. Версія модуля Arduino може бути зібрана в ручну, а деякі навіть готові модулі коштують менше 5 доларів;

- багатоплатформовність - програмне забезпечення Arduino працює під ОС Windows, Macintosh OSX і Linux. Більшість мікроконтролерів обмежується ОС Windows;

- проста і зрозуміла середу програмування - середа Arduino підходить як для початківців користувачів, так і для досвідчених. Arduino заснована на середовищі програмування Processing, що дуже зручно для викладачів, так як студенти працюють з цим середовищем будуть знайомі і з Arduino;

- програмне забезпечення з можливістю розширення і відкритим вихідним текстом - ПО Arduino випускається як інструмент, який може бути доповнений досвідченими користувачами. Мова може доповнюватися бібліотеками C++. Користувачі, які бажають зрозуміти технічні нюанси, мають можливість перейти на мову AVR C на якому заснований C++ [82]. Відповідно, є можливість додати код з середовища AVR-C в програму Arduino.

Характеристики Arduino UNO:

- мікроконтролер ATmega328;
- робоча напруга 5В;
- вхідна напруга (рекомендована) 7-12 В;
- вхідна напруга (граничне) 6-20 В;
- цифрові Входи/Виходи 14 (6 з яких можуть використовуватися як виходи ШІМ);
- аналогові входи 6;
- постійний струм через вхід / вихід 40 мА;
- постійний струм для виведення 3.3 В 50 мА;
- флеш пам`ять 32 Кб (ATmega328) з яких 0.5 Кб використовуються для завантажувача;
- ОЗУ 2 Кб (ATmega328);
- EEPROM 1 Кб (ATmega328);
- тактова частота 16 МГц [83].



Завдяки цим перевагам і було обрано Arduino UNO, який зображено на рисунку 3.5



Рисунок 3.5 – Плата Arduino UNO

### **3.2 Розробка клієнта з удосконаленою бібліотекою для системи «Розумний дім»**

Під час розробки клієнта було використано мову програмування JavaScript, так є це дозволить встановлювати клієнт не тільки на операційну систему Android, а також на інші операційні системи. Також це рішення дозволить зменшити час обробки даних при розпізнаванні голосових команд. Найбільшою перевагою є те що додаток буде легко масштабованим, та займати малу кількість простору на смартфоні.

Якщо розробляється легкий веб-додаток (або просто веб-сторінка) як частину клієнтської програми, то це краще зробити за допомогою WebView. Клас WebView - це розширення класу View для Android, який дозволяє відобразити веб-сторінки як частину макета додатку. Вона не включає жодних функцій повністю розробленого веб-переглядача, наприклад, елементів керування

навігацією або адресного рядка. Все, що робить WebView за замовчуванням, - це показ веб-сторінки.

Загальний сценарій, коли WebView корисно використовувати, - це коли потрібно надавати інформацію в програмі, яка може знадобитися для перегляду, наприклад, угода кінцевого користувача або керівництво користувача та інша корисна інформація.

Інший сценарій, за допомогою якого WebView може допомогти, полягає в тому, якщо ваша програма надає дані користувачеві, для якого завжди потрібне підключення до Інтернету, щоб отримувати дані, наприклад, данні з веб-сервісу.

У цьому випадку ви можете виявити, що в додатку Android, простіше створювати веб-переглядач, замість того, щоб виконувати мережевий запит, а потім обробляти дані та відтворювати їх у макеті Android. Замість цього ви можете створити веб-сторінку, розроблену для пристроїв Android, а потім застосувати WebView до вашої програми Android, яка завантажує веб-сторінку [84].

Завдяки цій технології і було розроблено веб-клієнт для смартфона, його вигляд зображено на рисунку 3.6.

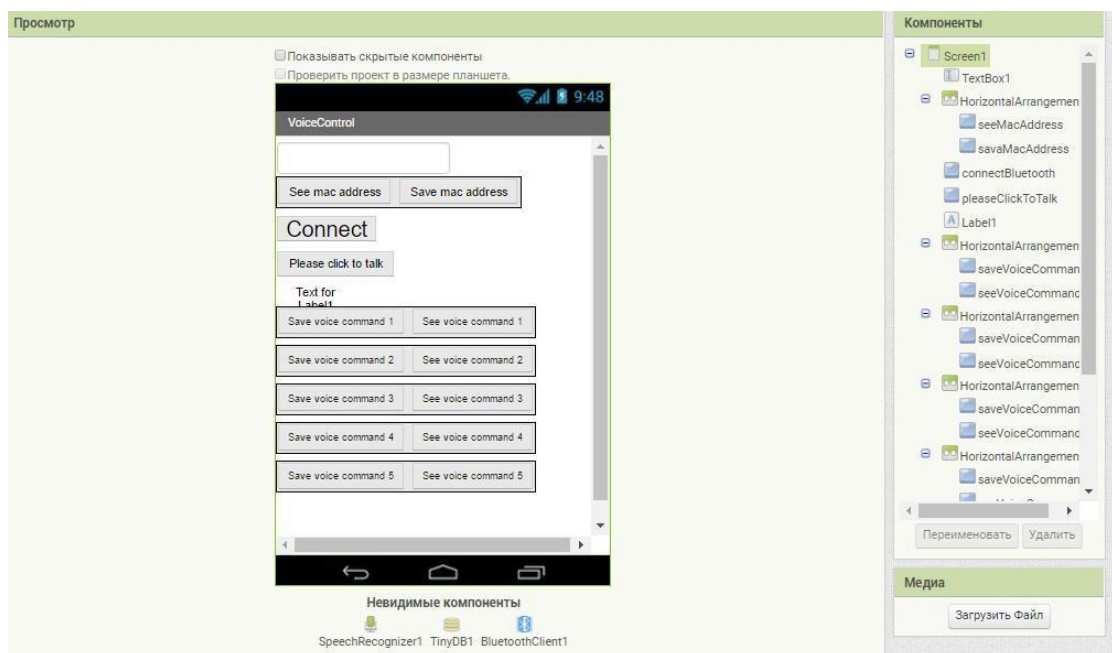


Рисунок 3.6 – Розробка веб-клієнту для смартфона

На рисунку 3.7 – відображено програмний код, який виконує запит до веб-сервісу, та отримує Token (або ticket), для подальших дій додатку. Якщо запит вдалий, то отримуємо у консоль відлагодження результат у вигляді ім'я користувача та Token (або ticket), який згенерував веб-сервіс.

Програмний код додатку винесено в додаток Б.

```
// Упаковщик в формат JSON
var serializer = new JavaScriptSerializer();

// Клиент веб-сервиса
var webClient = new ServiceReference1.WebServiceSoapClient();
WSHttpBinding binding = new WSHttpBinding();
binding.SendTimeout = new TimeSpan(0, 10, 0);

// Выполнить проверку логина и пароля
var loginRequestJson = webClient.LoginEx(User.login, User.password);
// Распаковать строку с результатами аутентификации
var loginRequest = (Dictionary<string, object>)serializer.Deserialize(loginRequestJson, typeof(Dictionary<string, object>));
// Успешность аутентификации
bool success = (bool)loginRequest["Success"];
Console.WriteLine(success);
// Имя пользователя
string userName = (string)loginRequest["UserName"];
Console.WriteLine(userName);
// Временный билет
string ticket = (string)loginRequest["Ticket"];
Console.WriteLine(ticket);
```

Рисунок 3.7 – Запит на отримання Token(або ticket), від веб-сервісу

### 3.3 Розробка веб-сервісу для системи «Розумний дім»

Підчас огляду сучасних мов програмування в першому розділу було обрано мову C# та фреймворк ASP.NET Core, для розробки веб-сервісу. Веб-сервіс відповідає усім сучасними заходами безпеки, а саме систему генерування Token, та HTTPS (рис. 3.8).

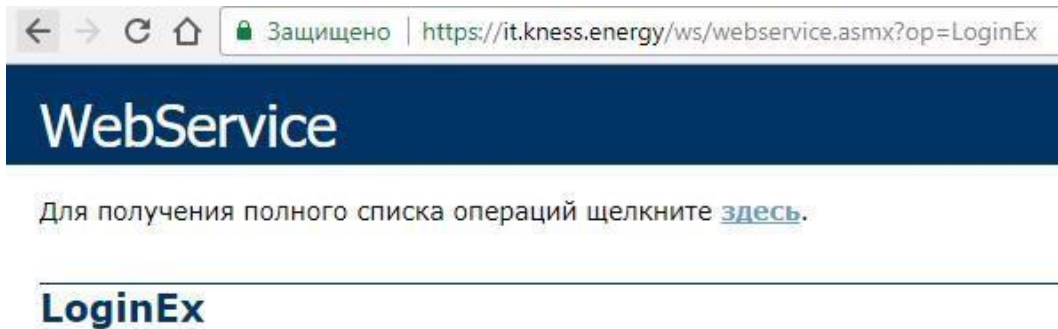


Рисунок 3.8 – HTTPS з'єднання з веб-сервісом

Для прикладу на рисунках 3.9-3.10 відображено http-get та http-post запити та відповіді до веб-сервісу.

### HTTP GET

В наступному прикладі показано HTTP-запит GET і відповідь на нього. Замість :

```
GET /ws/webservice.asmx/LoginEx?login=string&password=string HTTP/1.1
Host: it.kness.energy
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

Рисунок 3.9 – HTTP-GET запит та відповідь

### HTTP POST

В наступному прикладі показано HTTP-запит POST і відповідь на нього.

```
POST /ws/webservice.asmx/LoginEx HTTP/1.1
Host: it.kness.energy
Content-Type: application/x-www-form-urlencoded
Content-Length: length

login=string&password=string
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

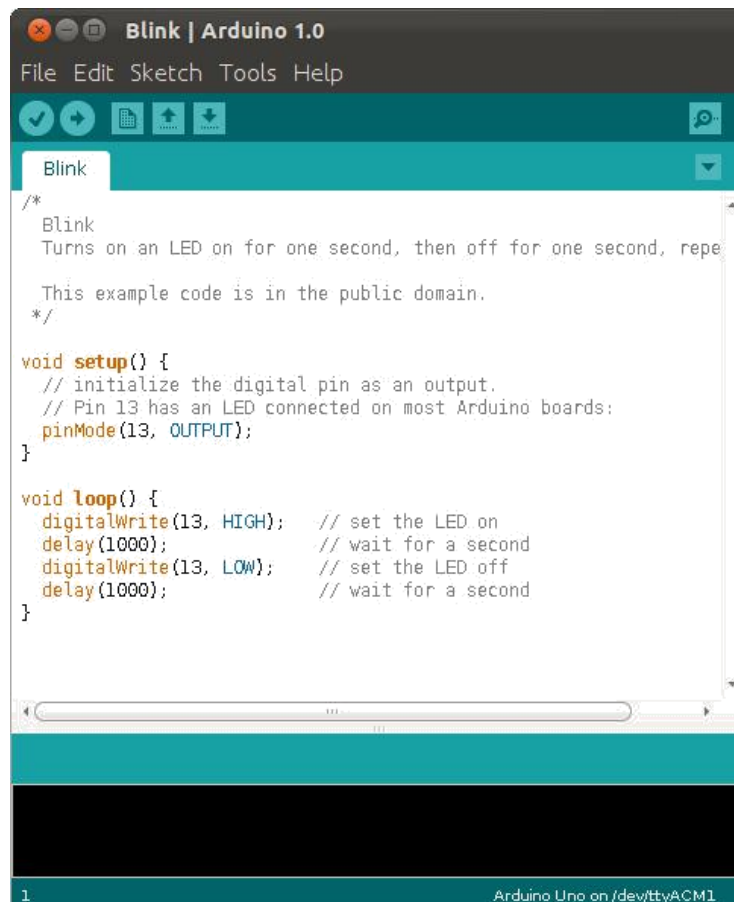
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

Рисунок 3.10 – HTTP-POST запит та відповідь

В додатку В написано лістинг коду.

### 3.4 Розробка програмного забезпечення для контролера в Arduino IDE

Arduino IDE – середовище розробки для програмування контролерів, зображено на рисунку 3.11. Програмування контролера здійснюється за допомоги C подібного синтаксису [85]. Програмне забезпечення включає в себе один файл, з чотирма методами, а саме: попередні налаштування, без кінцевий цикл, передача даних на веб-сервіс та управління за відхиленням. А також до цього файлу підключені стандартні бібліотеки, які дають змогу працювати за датчиками, реле та Ethernet-приймачем [86]. Лістинг винесено в додаток Б.



```
Arduino IDE - Blink | Arduino 1.0
File Edit Sketch Tools Help
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);            // wait for a second
}

1 Arduino Uno on /dev/ttyACM1
```

Рисунок 3.11 – Середовище розробки Arduino IDE

### 3.5 Тестування системи «Розумний дім»

Спочатку протестуємо клієнт разом з удосконаленою бібліотекою для розпізнавання голосових команд. Для зручності тестування будемо використовувати localhost, та порт 56247. Як бачимо на рисунку 3.12, сказана фраза «Мама мыла раму», була цілком розпізнана. Також у консолі було відображення того, як алгоритм наближався до вірного рішення.

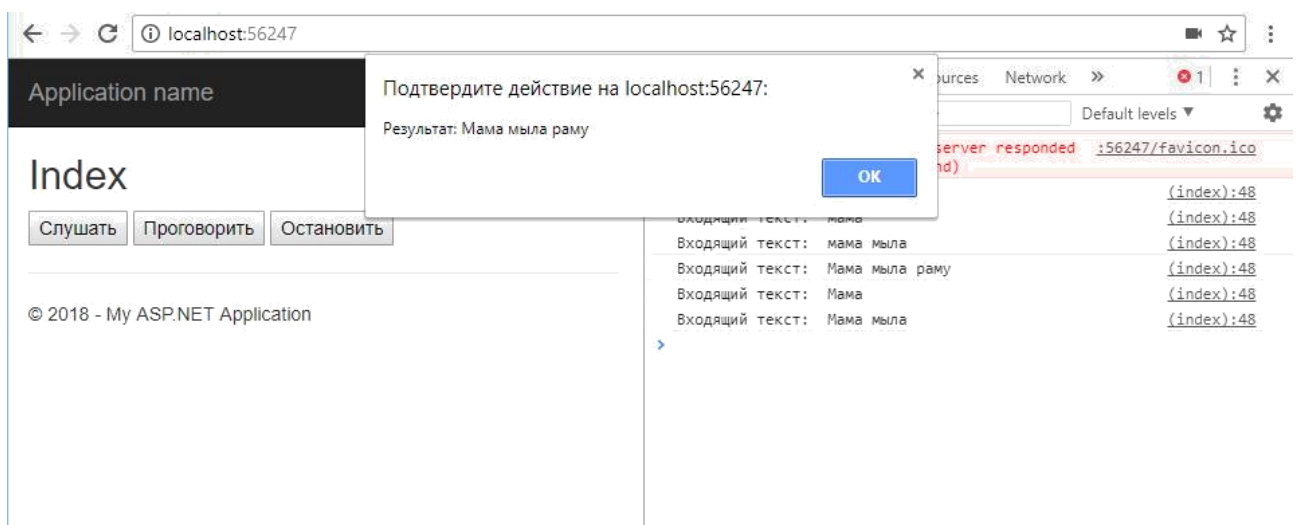


Рисунок 3.12 – Результати виводу клієнта

Наступним кроком перевіримо веб-сервіс. Для цього зайдемо на стартову сторінку веб-сервісу, та з переліку готових функцій виберемо авторизацію (рис. 3.13). Введемо логін та пароль, та натиснемо «Запуск».

# WebService

Для получения полного списка операций щелкните [здесь](#).

## LoginEx

### Тест

Чтобы протестировать операцию с использованием HTTP-протокола POST, нажми

Параметр	Значение
login:	PIDVACHETSKYI_DA
password:	<input type="password"/>

### SOAP 1.1

В следующем примере показаны запрос и ответ SOAP 1.1. Вместо элементов-за

```
POST /ws/webservice.asmx HTTP/1.1
```

Рисунок 3.13 – Введення логіна та пароля для авторизації

У результаті, якщо логін та пароль є правильними, отримуємо назву користувача, та Token, саме це зображено на рисунку 3.14.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<string xmlns="http://tempuri.org/">
  {"Success":true,"Id":"PIDVACHETS","UserName":"Підвашецький Д.А.,"Ticket":"840EBCEE-F8DE-4D4F-969C-
  007C32074831","FailReason":null,"NeedChangePassword":false,"TempPasswordRequired":false,"TempPasswordMessage":null}
</string>
```

Рисунок 3.14 – Вдала авторизація

Останнім кроком перевіримо роботу мікроконтролера Arduino. Перевіримо давач температури та вологості, управління за відхиленням, та спрацювання спеціального реле . Якщо температура у приміщенні перевищує 27 градусів за Цельсієм у нас вмикається система охолодження приміщення. Якщо температура

становить меньше 27 градусів і нище – система охолодження не вмикається.  
Результат зображений на рисунках 3.15 та 3.16.

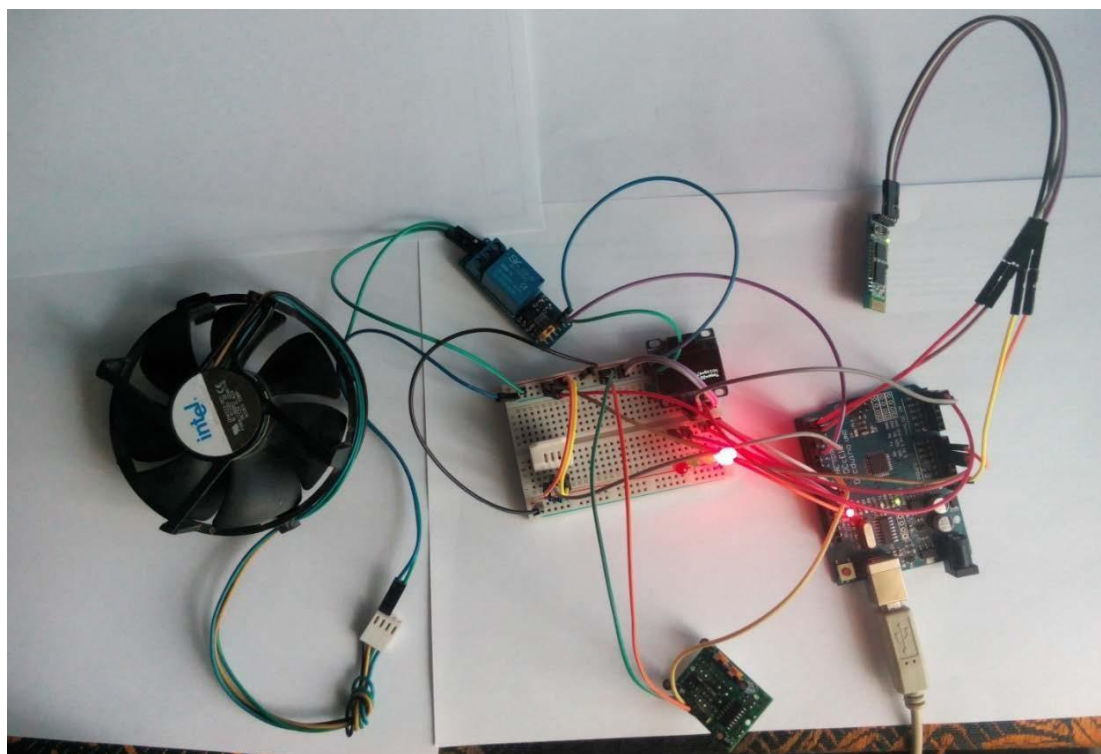


Рисунок 3.15 – Результат при температурі 26.7 градусів



Рисунок 3.16 – Відображення поточної температури і вологості



## ВИСНОВКИ

В ході виконання кваліфікаційної роботи бакалавра було проаналізовано різні системи розумних будинків та розроблена комп'ютеризована системи управління «Розумний дім», за допомогою якої жителі дома можуть спостерігати як система корегує навколишні зміни, або за допомоги голосових команд керувати будинком. Програма має зручний та зрозумілий інтерфейс для користувача.

В першому розділі був проведений аналіз об'єкту автоматизації, розглянуті сучасні комп'ютеризовані системи управління розумним будинком. Зроблена постановка задачі.

В другому розділі для даної системи було проведено аналіз функцій, та на його основі побудована UML-діаграма прецедентів. Було розроблено структурну та функціональну схему даної системи. Було розроблено UML-діаграми діяльності та послідовності.

В третьому розділі було проаналізовано та підібрані механічні засоби для роботи з системою, розроблено програмне забезпечення для клієнта, веб-сервісу та мікроконтролеру на базі плати Arduino.

Після проведення тестування можна зробити висновок, що розроблена система може бути використана на сучасних розумних будинках.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Підвашецький Д. А. Удосконалення алгоритму розпізнавання голосових команд для управління комп'ютеризованою системою «розумний будинок». : [Електронний ресурс]: // ВНТУ – Режим доступу до ресурсу: <http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/17446/3135.pdf?sequence=3>.

2. Підвашецький Д. А. Розробка комп'ютеризованої системи управління «розумний будинок» : [Електронний ресурс]: // ВНТУ – Режим доступу до ресурсу: <http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10971/1037.pdf?sequence=3>.

3. Підвашецький Д. А. Вимірювання концентрації промислово-логічного контролера VIPA 200. : [Електронний ресурс]: // ВНТУ – Режим доступу до ресурсу: <http://ir.lib.vntu.edu.ua/handle/123456789/14809>.

4. Розумний дім: [Електронний ресурс]: // Wikipedia – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D1%83%D0%BC%D0%BD%D0%B8%D0%B9\\_%D0%B4%D1%96%D0%BC](https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D1%83%D0%BC%D0%BD%D0%B8%D0%B9_%D0%B4%D1%96%D0%BC).

5. Розумний дім: [Електронний ресурс]: // Sitem – Режим доступу до ресурсу: <http://sitem.com.ua/>.

6. Система розумний дім: [Електронний ресурс]: // Ecotown – Режим доступу до ресурсу: <https://ecotown.com.ua/news/Systema-Rozumnyy-dim-zmenschuye-vytraty-na-komposluhy-do-30/>.

7. Програмируемый выключатель света: [Електронний ресурс]: // Geektimes – Режим доступу до ресурсу: <https://geektimes.ru>.

8. Історія розумний дім: [Електронний ресурс]: // wiki – Режим доступу до ресурсу: [http://wiki.tntu.edu.ua/%D0%A0%D0%BE%D0%B7%D1%83%D0%BC%D0%BD%D0%B8%D0%B9\\_%D0%B4%D1%96%D0%BC](http://wiki.tntu.edu.ua/%D0%A0%D0%BE%D0%B7%D1%83%D0%BC%D0%BD%D0%B8%D0%B9_%D0%B4%D1%96%D0%BC).

9. Функції та комфорт розумного дому: [Електронний ресурс]: // melask – Режим доступу до ресурсу: <http://melask.com.ua/rozumniy-dim/komfort.html>.
10. Розумний дім – технологія економії: [Електронний ресурс]: // right-choice – Режим доступу до ресурсу: <http://right-choice.com.ua/rozumniy-dim-tehnologiya-ekonomiyi-zruchnosti-i-komfortu-visokogo-rivnya>.
11. Розумний будинок: [Електронний ресурс]: // lady.tochka – Режим доступу до ресурсу: <http://lady.tochka.net/ua/58959-hto-takoe-umnyu-dom/>.
12. Керований пристрій «Розумного» будинку: [Електронний ресурс]: // sutem – Режим доступу до ресурсу: <http://sutem.com.ua/7113smartbus.php>.
13. Зовнішній датчик руху «Розумного» будинку: [Електронний ресурс]:sutem – Режим доступу до ресурсу: <http://sutem.com.ua/7135smartbus.php>.
14. Шлюзи зв'язку «Розумного» будинку: [Електронний ресурс]: // sutem – Режим доступу до ресурсу: <http://sutem.com.ua/7141smartbus.php>.
15. Керуючі пристрої «Розумного» будинку: [Електронний ресурс]: // sutem – Режим доступу до ресурсу: <http://sutem.com.ua/7121smartbus.php>.
16. Переваги «Розумного» будинку: [Електронний ресурс]: // klyuch – Режим доступу до ресурсу: <http://klyuch.com.ua/m/articles/economy/yaki-perevagy-rozumnogo-domu-/>.
17. Delaney B, Jayant N, Hans M. A Low-power , Fixed-point Front-end Feature Extraction for a Distributed Speech Recognition System[J]. HP Laboratories Technical Report, 2001, 26(9): 252-254.
18. Christophe L, Georges L, Nocera P. Reducing Computational and Memory Cost for Cellular Phone Embedded Speech Recognition System[J] . Proceedings of the IEEE, 2012, 85(9): 112-115.
19. Douglas A, Richard C. Robust text-independent speaker identification using Gaussian mixture speaker models [J]. IEEE Trans Speech and Audio Processing, 1995, 3(1): 77-80.
20. Deller John R, Proakis John G, Hansen John H L. Discrete-Time Processing of Speech Signals [M]. Macmillan Publishing Company, 2003.

21. Reynolds D A, Quatieri T F, Dunn R B. Speaker verification using adapted gaussian mixture models [J]. *Digital Signal Processing*, 2000(10): 19-41.
22. Yang Hongwu, HuangDezhi, Cai Lian-hong. Perceptually Weighted Mel-Cepstrum Analysis of Speech Based Psychoacoustic Model[J]. *IEICE TRANS. INF.SYST*, 2006, E89-D (12): 1-4.
23. L. R. Bahl, P. Gopalakrishnan, and R. L. Mercer. Search issues in large vocabulary speech recognition. In *Proceedings of the 1993 IEEE Workshop on Automatic Speech Recognition*, Snowbird, UT, 1993.
24. B´anhalmi, A. Kocsor, and R. Busa-Fekete. Counter-example generation-based one-class classification. In *Proceedings of ECML*, pages 543–550, 2007.
25. J. Dombi. Towards a general class of operators for fuzzy systems. *IEEE Transaction on Fuzzy Systems*, 16(2):477–484, 2008.
26. D. Dubois and H. Prade. *Fundamentals of Fuzzy Sets*. Kluwer Academic Publisher, 2000.
27. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. WileySons, New York, 1973.
28. W. Gerstner and W. M. Kistler. *Spiking Neuron Models*. Cambridge University Press, 2002.
29. J. R. Glass. A probabilistic framework for segment-based speech recognition. *Computer Speech and Language*, 17(2):137–152, 2003.
30. G. Gordos and G. Tak´acs. *Digital Speech Processing (in Hungarian)*. Muszaki " K¨onyvkiad´o, 1983.
31. G. Gosztolya, A. B´anhalmi, and L. T´oth. Using one-class classification techniques in the antiphoneme problem. In *Proceedings of the 2009 Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, volume LNCS 5524, pages 433–440, Porto, Portugal, 2009.
32. G. Gosztolya, J. Dombi, and A. Kocsor. Applying the Generalized Dombi Operator family to the speech recognition task. *Journal of Computing and Information Technology*, 17(3):285–293, 2009.

33. G. Gosztolya and A. Kocsor. Improving the multi-stack decoding algorithm in a segment-based speech recognizer. In Proceedings of the 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE), volume LNCS 2718, pages 744–749, Loughborough, England, UK, 2003.
34. G. Gosztolya and A. Kocsor. Aggregation operators and hypothesis space reductions in speech recognition. In Proceedings of the 2004 Conference on Text, Speech and Dialogue (TSD), volume LNCS 3206, pages 315–322, Brno, Czech Republic, 2004.
35. G. Gosztolya and A. Kocsor. A hierarchical evaluation methodology in speech recognition. *Acta Cybernetica*, 17(2):213–224, 2005.
36. G. Gosztolya and A. Kocsor. Speeding up dynamic search methods in speech recognition. In Proceedings of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE), volume LNCS 3533, pages 98–100, Bari, Italy, 2005.
37. T. N. Sainath. Acoustic landmark detection and segmentation using the McAulay-Quatieri sinusoidal model. Master’s thesis, MIT, 2005.
38. R. Schwartz, L. Nguyen, and J. Makhoul. Multiple-pass Search Strategies, chapter 18, pages 429–456. Kluwer Academic Publisher, Philadelphia, PA, 1996.
39. D. M. Tax. One-class classification; Concept-learning in the absence of counter-examples. PhD thesis, Delft University of Technology, 2001.
40. L. T’oth, A. Kocsor, and J. Csirik. On Naive Bayes in speech recognition. *International Journal of Applied Mathematics and Computer Science*, 15(2):287–294, 2005.
41. L. T’oth, A. Kocsor, and G. Gosztolya. Telephone speech recognition via the combination of knowledge sources in a segmental speech model. *Acta Cybernetica*, 16(4):643–657, 2004.
42. S. Young. Statistical modelling in continuous speech recognition. In Proceedings of the International Conference on Uncertainty in Artificial Intelligence, Seattle, 2001.

43. R. S. Ransing and M. Rajput, "Smart home for elderly care, based on Wireless Sensor Network," Nascent Technologies in the Engineering Field (ICNTE), 2015 International Conference on, Navi Mumbai, 2015, pp. 1-5.

44. M. M. A. Jamil and M. S. Ahmad, "A pilot study: Development of home automation system via raspberry Pi," Biomedical Engineering (ICoBE), 2015 2nd International Conference on, Penang, 2015, pp. 1-4.

45. C. Felix and I. Jacob Raglend, "Home automation using GSM," Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on, Thuckafay, 2011, pp. 15-19.

46. R. A. Ramlee, M. A. Othman, M. H. Leong, M. M. Ismail and S. S. S. Ranjit, "Smart home system using android application," Information and Communication Technology (ICoICT), 2013 International Conference of, Bandung, 2013, pp. 277-280.

47. N. Dickey, D. Banks and S. Sukittanon, "Home automation using Cloud Network and mobile devices," Southeastcon, 2012 Proceedings of IEEE, Orlando, FL, 2012, pp. 1-4.

48. S. Folea, D. Bordenca, C. Hotea and H. Valean, "Smart home automation system using Wi-Fi low power devices," Automation Quality and Testing Robotics (AQTR), 2012 IEEE International Conference on, Cluj-Napoca, 2012, pp. 569-574.

49. M. A. Ullah, A. R. Celik, "An Effective Approach to Build Smart Building Based on Internet of Things (IoT)", Journal of Basic and Applied Scientific Research, issues 6, 2016, pp. 56-62.

50. P. S. Chinchansure and C. V. Kulkarni, "Home automation system based on FPGA and GSM," Computer Communication and Informatics (ICCCI), 2014 International Conference on, Coimbatore, 2014, pp. 1-5.

51. J. Han, C. S. Choi, W. K. Park, I. Lee and S. H. Kim, "Smart home energy management system including renewable energy based on ZigBee and PLC," 2014 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2014, pp. 544-545.

52. Інтелектуальні системи освітлення: [Електронний ресурс]: // dss-bi – Режим доступу до ресурсу: <http://dss-bi.com.ua/sitelab1/>.

53. Логічні пристрої «Розумного» будинку: [Електронний ресурс]: // sitem – Режим доступу до ресурсу: <http://sitem.com.ua/7smartbus.php>.
54. Схеми керування апаратурою «Розумного» будинку: [Електронний ресурс]: // sitem – Режим доступу до ресурсу: <http://sitem.com.ua/724smartbus.php>.
55. Освітлення «Розумного» будинку: [Електронний ресурс]: // economstroy – Режим доступу до ресурсу: <http://economstroy.com.ua/mvoprosyiotvety/2302-rozumnybudunokosvitlena.html>.
56. Гололобов В. Н. «Умный дом» своими руками. / В.Н Гололобов. М.:НТ
57. Датчики температуры. Комплексные поставки. Разработчик: группа предприятий Метран. Тематический каталог. 2015.
58. Сопер М. Э. Практические советы и решения по созданию «Умного дома» / М. Э. Сопер – М.: НТ Пресс, 2012. – 432 с.
59. Тесля Е. А. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире / Е. А. Тесля. – Санкт Петербург, 2014. – 224с.
60. Харке В. Н. «Умный дом. Объединение в сеть бытовой техники и систем коммуникаций в жилищном строительстве» / В. Н. Харке– М.: Техносфера, 2013. – 292с.
61. Элсенпитер Т. Р. «Умный Дом строим сами» / Т. Р. Элсенпитер, Велт Дж / КУДИЦ-ОБРАЗ. 2015. – 384с.
62. Гололобов В. Н. «Умный дом» своими руками. / В. Н. Гололобов – М.: НТ Пресс, 2016. – 416 с.
63. Авдеевский А. В. «Крыша для интеллекта» - / А. В. Авдеевский «Журнал сетевых решений LAN», №12 2012 г.
64. Алексеев Г. П. Электромонтаж и наладка системы «Умный дом». Руководство по выполнению базовых экспериментов. - / Г. П. Алексеев ЭМНСУД.001 РБЭ (997)- Челябинск: ИПЦ «Учебная техника», 2012. –223с.
65. Королев Ю. Ф. «УМНЫЙ ДОМ: приятная неизбежность» - / Ю. Ф. Королев Ю.
66. Харке В. Н. «Умный дом» - / В. Н. Харке Техносфера, Москва 2010 г.

67. Поляк Б. Т. Робастная устойчивость и управление / Б. Т. Поляк, П. С. Щербаков / - М.: Наука, 2012.- 169 с.
68. Попов Е. П. Приближенные методы исследования нелинейных автоматических систем / Е. П. Попов, И. П. Пальтов / Физматгиз, М., 2014.- 524 с.
69. Рагазин Д. А. Синтез релейных самонастраивающихся систем с заданными показателями качества Текст. /В.Е. Вохрышев, И.А. Капустин, Д.А. Рагазин, //Вестник Самарскогогос. техн. ун-та. Сер. Техн. науки -2017. -№ 1(23).- 22-26.
70. Сидоров А. Н. Исследование нелинейных систем в случае установления двухчастотных процессов/ А. Н. Сидоров, И. П. Коротаева // Техническая кибернетика, 2015. №5, 148-158 с.
71. Уоллес В. А. Основы программирования для «чайников». / Уоллес Вонг – М.:Диалектика, 2012. – 280 с.
72. Балдин К.В., Уткин В.Б. Информационные системы. – М.: Дашков,2016. – 395 с.
73. Гуджоян О.Л. Методы принятия управленческих решений. – М.: 2012.
74. Левин Б.Р. Вероятностные модели и методы в системах связи и управления/Левин Б. – М.: Радио и связь, 2015.– 312 с.
75. Мартин Ф. UML. Основы/Мартин Фаулер. – М.: Символ-Плюс, 2012. – 432 с.
76. Лафоре Р. Объектно–ориентированное программирование в C++. 4 – издание / Лафоре Р. – М.: Санкт–Петербург, 2014. – 923 с.
77. Пратт Т. Языки программирования: разработка и реализация/ Зелкович М. – Спб.: Питер, 2012. – 688 с.
78. Элементы UML [Электронный ресурс]: // KDE – Режим доступа: <http://docs.kde.org/stable/uk/kdesdk/umbrello/uml-elements.html>.
79. ГОСТ 19.101 - 77 – Розробка програмної документації, видів програм та програмних документів. – М.: Госстандарт, 1981.
80. ГОСТ 19.503 – 79 – Керівництво системного програміста. Вимоги до змісту та оформлення. – М.: Госстандарт, 1980р.



81. ГОСТ 19.106 - 78 – Вимоги до програмних документів, виконані друкованим способом. – М.: Госстандарт, 1982р.

82. ГОСТ 19.505 – 79 – Керівництво оператора. Вимоги до змісту та оформлення. – М.: Госстандарт, 1980р.

83. Кавецький В. В. Обґрунтування інноваційних рішень. Практикум : [навч. посіб.] / В. В. Кавецький, В. О. Козловський, І. В. Причепка – Вінниця : ВНТУ, 2013. – 113 с.

84. Липаев В.В. Обеспечение качества программных средств./В.В. Липаев.—Москва: Синтег, 2011.-380с., 28 илл.

85. Благодатских В.А. Стандартизация разработки программных средств: Учеб. пособие под ред. О.С.Разумова. М./ В.А. Волнин, В.А. К.Ф. Посакалов. - Финансы и статистика, 2013. 284 с.

86. Официальный русскоязычный сайт платформы Arduino [електронний ресурс] – Режим доступу: <http://arduino.ru/>

87. Черкесов Г. Н. Надежность аппаратно-программных комплексов/Г.Н. Черкесов. - СПб.: «Питер», 2015. 479с.

88. Традиційні інформаційні системи та автоматизовані: [Електронний ресурс] : // flybb – Режим доступу до ресурсу: <http://ism.flybb.ru/topic83.html>.

89. Черникова П.Д. Технические расчеты и обоснования в дипломных проектах/ П.Д. Черникова.- Минск 2003.-188с.

## ДОДАТОК А

### Лістинг програм

#### Лістинг А.1 - Лістинг програмного забезпечення

```

using System;
using System.Collections.Generic;
using System.ComponentModel;

using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;

using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.IO;

namespace arduino
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            notifyIcon1.Visible = false;
            this.notifyIcon1.MouseClick += new
            MouseEventHandler(notifyIcon1_MouseClick);
            this.Resize += new
            System.EventHandler(this.Form1_Resize);

            //    Открываем порт, и задаем скорость в 9600 бод
        }
        //***** поток ком порта

        private void serialPort1_DataReceived(object sender,
        System.IO.Ports.SerialDataReceivedEventArgs e)
        {
            string vlag = serialPort1.ReadLine();
            this.BeginInvoke(new LineReceivedEvent (LineReceived), vlag);
        }
        private void serialPort1_DataReceived1(object sender,
        System.IO.Ports.SerialDataReceivedEventArgs e)
    }
}

```

```

    {
        string temp = serialPort1.ReadLine();

        this.BeginInvoke(new LineReceivedEvent1(LineReceived1),
temp);
    }
    private void serialPort1_DataReceived2(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)

    {

        string svet = serialPort1.ReadLine();
        this.BeginInvoke(new LineReceivedEvent2(LineReceived2),
svet);
    }
    private void serialPort1_DataReceived3(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)

    {
        string mowe = serialPort1.ReadLine();
        this.BeginInvoke(new LineReceivedEvent3(LineReceived3),
mowe);
    }
    private void serialPort1_DataReceived4(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)

    {
        string relestats = serialPort1.ReadLine();
        this.BeginInvoke(new LineReceivedEvent4(LineReceived4),
relestats);
    }

    //запись влажности
    private delegate void LineReceivedEvent(string vlag);
    private void LineReceived(string vlag)
    {

        textBox1.Text = vlag;
        string path = "График_влажности.txt";
        string date = DateTime.Now.ToString();
        // Создание файла и запись в него

        using (StreamWriter sw = File.AppendText(path))
        {
            sw.WriteLine(vlag);

            sw.WriteLine(date);
        }
    }

    //запись температуры
    private delegate void LineReceivedEvent1(string temp);

```

```

private void LineReceived1(string temp)
{
    textBox2.Text = temp.ToString();
    string path = "График_температуры.txt";
    string date = DateTime.Now.ToString();
    //    Создание файла и запись в него

    using (StreamWriter sw = File.AppendText(path))
    {
        sw.WriteLine(temp);
        sw.WriteLine(date);
    }
}
//    запись света
private delegate void LineReceivedEvent2(string svet);

private void LineReceived2(string svet)
{
    if (Convert.ToInt32(svet) > 100 && Convert.ToInt32(svet) <
200)
    {
        textBox3.Text = "Хорошее освещение";
    }
    else if (Convert.ToInt32(svet) < 100)
    {
        textBox3.Text = "Мало освещения";
    }
    else
    {
        textBox3.Text = "Достаточно освещения";
    }
    string path = "График_sveta.txt";
    string date = DateTime.Now.ToString();
    //    Создание файла и запись в него

    using (StreamWriter sw = File.AppendText(path))
    {
        sw.WriteLine(svet);
        sw.WriteLine(date);
    }
}
// move
private delegate void LineReceivedEvent3(string move);

private void LineReceived3(string move)
{

```

```

        if
(Convert.ToInt32(move) ==
        1)
        textBox4.Text =
            "Движение";

else
    textBox4.Text = "Нет_Движения";
string path = "График_move.txt";
string date = DateTime.Now.ToString();

    //    Создание файла и запись в него
using (StreamWriter sw = File.AppendText(path))
{
    sw.WriteLine(textBox4.Text + " " + date);
}

}
// rele

private delegate void LineReceivedEvent4(string rele);
private void LineReceived4(string rele)
{
    if (Convert.ToInt32(rele) == 1)

        textBox5.Text = "Включено";
    else
textBoх5.Text = "Выключено",
        string path = "График_rele.txt";

        string date = DateTime.Now.ToString();
        //    Создание файла и запись в него
using (StreamWriter sw = File.AppendText(path))
{
        sw.WriteLine(textBox5.Text+" "+date);
    }
}
private void Form1_Resize(object sender, EventArgs e)

{
    if (this.WindowState == FormWindowState.Minimized)
    {
        this.Hide();

        notifyIcon1.Visible = true;
    }
}
private void notifyIcon1_MouseClick(object sender,
MouseEventArgs e)

{

```

```

    this.Show();
    this.WindowState = FormWindowState.Normal;
    notifyIcon1.Visible = false;

}
private void label1_Click(object sender, EventArgs e)
{

}
private void демоToolStripMenuItem_Click(object sender,
EventArgs e)
{
    panel1.Visible = false;

    tabControl1.Visible = true;
    panel2.Visible = false;
    chart1.ChartAreas[0].AxisX.ScaleView.Zoom(0, 30);
    chart1.ChartAreas[0].CursorX.IsUserEnabled = true;

    chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
    chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
    chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside =
true;
    StreamReader streamReader = new
StreamReader("График_температуры.txt");
    chart1.Series[0].Points.Clear();

    while (!streamReader.EndOfStream)
    {
        string Y = streamReader.ReadLine();
        string X = streamReader.ReadLine();

        chart1.Series[0].Color = Color.Red;
        chart1.Series[0].BorderWidth = 1;
        chart1.Series[0].Points.AddXY(X, Y);

    }
    streamReader.Close();

    //График_влажности

    chart2.ChartAreas[0].AxisX.ScaleView.Zoom(0, 30);
    chart2.ChartAreas[0].CursorX.IsUserEnabled = true;
    chart2.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
    chart2.ChartAreas[0].AxisX.ScaleView.Zoomable = true;

    chart2.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside =
true;
    streamReader = new StreamReader("График_влажности.txt");
    chart2.Series[0].Points.Clear();
    while (!streamReader.EndOfStream)

    {

```

```

        string Y = streamReader.ReadLine();
        string X = streamReader.ReadLine();

        chart2.Series[0].Color = Color.Red;
        chart2.Series[0].BorderWidth = 1;
        chart2.Series[0].Points.AddXY(X, Y);
    }

    streamReader.Close();

    //График_sveta
    chart3.ChartAreas[0].AxisX.ScaleView.Zoom(0, 30);
    chart3.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
    chart3.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
    chart3.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
    chart3.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = true; streamReader
    = new
    StreamReader("График_sveta.txt");
    chart3.Series[0].Points.Clear();

    while (!streamReader.EndOfStream)
    {
        string Y = streamReader.ReadLine();

        string X = streamReader.ReadLine();

        chart3.Series[0].Color = Color.Red;
        chart3.Series[0].BorderWidth = 1;

        chart3.Series[0].Points.AddXY(X, Y);
    }
    streamReader.Close();
}

private void выходToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Application.Exit();
}

private void поточныйСтанToolStripMenuItem_Click(object
sender, EventArgs e)
{
    panel1.Visible = true;
    tabControl1.Visible = false;

    panel2.Visible = false;
}

```

```

private void textBox3_TextChanged(object sender, EventArgs e)
{
}
private void panell1_Paint(object sender, PaintEventArgs e)
{
}
private void Form1_Load(object sender, EventArgs e)
{
    panell1.Visible = true;
    tabControll1.Visible = false;
    panel2.Visible = false;
}

private void
состояниеРелеИДатчикаДвиженияToolStripMenuItem_Click(object
sender, EventArgs e)
{
    panell1.Visible = false;
    tabControll1.Visible = false;

    panel2.Visible = true;

    StreamReader streamReader = new
StreamReader("График_move.txt"); DataSet
ds = new DataSet();

    ds.Tables.Add("Score");
    string header = streamReader.ReadLine();
    string[] col =
System.Text.RegularExpressions.Regex.Split
(header, " "); for (int c = 0; c <
col.Length; c++)
    {
        ds.Tables[0].Columns.Add(col[c]);
    }
    string row = streamReader.ReadLine();

    while (row != null)
    {
        string[] value =
System.Text.RegularExpressions.Regex.Split
(row, " "); ds.Tables[0].Rows.Add(value);

        row = streamReader.ReadLine();
    }
    dataGridView2.DataSource = ds.Tables[0];
}

```



```

streamReader.Close();
streamReader = new StreamReader("График_rele.txt");
    ds = new DataSet();

    ds.Tables.Add("Score1");
    header = streamReader.ReadLine();
    col = System.Text.RegularExpressions.Regex.Split(header, "
");
    for (int c = 0; c < col.Length; c++)

    {
        ds.Tables["Score1"].Columns.Add(col[c]);
    }
    row = streamReader.ReadLine();

    while (row != null)
    {
        string[] value =
System.Text.RegularExpressions.Regex.Split(row, " ");
        ds.Tables["Score1"].Rows.Add(value);

        row = streamReader.ReadLine();
    }
    dataGridView1.DataSource = ds.Tables["Score1"];
    streamReader.Close();

}

private void проПрограммуToolStripMenuItem_Click(object
sender, EventArgs e)

{
    MessageBox.Show("Программа разработана для работы с
контроллером Arduino UNO, подключение при помощи порта COM3 и
позволяет отображать и" +

        "сохранять на компьютер данные про температуру, освещение,
влажность и состояние реле и движение ", "Про программу");

}

private void проАвтораToolStripMenuItem_Click(object sender,
EventArgs e)
{
    Form2 pro_avtora = new Form2();
    pro_avtora.Show();

}

private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
{

```

```
}  
  
private void button1_Click(object sender, EventArgs e)  
  
{  
    try  
    {  
        serialPort1.PortName = comboBox1.Text;  
  
        serialPort1.BaudRate = 9600;  
        serialPort1.DtrEnable = true;  
        serialPort1.Open();  
        serialPort1.DataReceived += serialPort1_DataReceived;  
  
        serialPort1.DataReceived += serialPort1_DataReceived1;  
        serialPort1.DataReceived += serialPort1_DataReceived2;  
        serialPort1.DataReceived += serialPort1_DataReceived3;  
        serialPort1.DataReceived += serialPort1_DataReceived4;  
  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show(ex.Message);  
    }  
}  
  
private void panel2_Paint(object sender, PaintEventArgs e)  
  
{  
  
}  
}  
}
```

## ДОДАТОК Б

### Програма завантаженої на контролер Arduino Uno

#### Лістинг Б.1 - Програма завантаженої на контролер Arduino Uno

```

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h> // Скачання бібліотека для дисплея.

#include <Adafruit_SSD1306.h> // Скачання бібліотека
для дисплея. https://yadi.sk/d/9F\_uWlwIZUDna #define
OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);
#include "DHT.h"

#define DHTPIN 2
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

int Relay = 4;
int sensePin=0;
int ledPin=9;
int pirPin = 7;

int minSecsBetweenEmails = 60; // 1 min
long lastSend = -minSecsBetweenEmails * 1000;
int rel = 0;
int mowe = 0;

void setup() {
  Serial.begin(9600);
  pinMode(pirPin, INPUT);

  pinMode(ledPin, OUTPUT);
  analogReference(DEFAULT);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  pinMode(Relay, OUTPUT);

  display.clearDisplay(); // Clear the buffer. / Очистить буфер.
  display.setTextColor(WHITE); // Цвет текста.
  display.setTextSize(1); // Размер текста (1)
  display.setCursor(0,0); // Устанавливаем курсор в колонку 0,
строку 0. на самом деле это строка №1, т.к. нумерация начинается с
0.

  display.println("Denis Podvashetskiy");
  display.setTextSize(1); // Размер текста (1)
  display.setCursor(40,10);

```

```

display.println("and");
display.setTextSize(1); // Размер текста (1)
display.setCursor(20,20);
display.println("SMART HOUSE");
display.setTextSize(3); // Размер текста (1)

display.setCursor(0,30);
display.println("PRESENT");
display.display();
dht.begin();

}
void sentParams(float temp,float vol, int svet , int mowe , int
relestats)
{

Serial.print(vol);
Serial.print("\t\n");
Serial.print(temp);
Serial.print("\t\n");

Serial.print(svet);
Serial.print("\t\n");
Serial.print(mowe);
Serial.print("\t\n");
Serial.print(relestats);
Serial.print("\n");
}

void loop() {
rel = 0;

mowe = 0;
float h = dht.readHumidity();
float t = dht.readTemperature();

if (isnan(t) || isnan(h)) {
}
else {
display.clearDisplay();

display.setTextSize(1); // Меняем размер текста (5).
display.setCursor(0,10); // Устанавливаем курсор в колонку 0,
строку 2. (строка №3 ).
display.print("Temp="); // Печатаем 3 строку.
display.println(t);

display.setCursor(0,20);
display.print("Volog="); // Печатаем 3 строку.
display.println(h);
display.display();

if(t>=27)

```

```

{
  digitalWrite(Relay, LOW);
  rel=1;

}
else
{
  digitalWrite(Relay, HIGH);

  rel=0;
}
}
int value = analogRead(sensePin);

value = constrain(value,600, 900);
int ledLevel = map (value, 900, 600, 255, 0 );
analogWrite(ledPin, ledLevel);
long now = millis();

if (digitalRead(pirPin) == HIGH)
{
  mowe = 1;
  if (now > (lastSend + minSecsBetweenEmails * 1000))
  {

    display.setTextSize(2); // Меняем размер текста (5).
    display.setCursor(15,30); // Устанавливаем курсор в колонку 0,
    строку 2. (строка №3 ).
    display.println("MOVEMENT");
    display.display();

    lastSend = now;
  }
}
sentParams (t,h,ledLevel,mowe,rel);

delay(1000);

}

```

## ДОДАТОК В ЛІСТИНГ ВЕБ-СЕРВІСУ

Лістинг В.1 - частковий лістинг веб-сервісу.

```

using System.IO;
using System.ServiceModel;
using System.ServiceModel.Channels;
using System.ServiceModel.Description;
using System.Xml;
using ITnet2.Common.Tools;
using ITnet2.Server.Session;
using ITnet2.Server.Session.Operations;

    /// <summary>
    /// Клас для вызова функцій на веб-сервісе SOAP/
    /// </summary>
    /// <typeparam name="TIn">Тип. описує входні
параметри функції (Request)</typeparam>
    /// <typeparam name="TOut">Тип. описує результат
функції (Response)</typeparam>
    /// <remarks>Создайте в VisualStudio ссылку на сервис
(http://...?wsdl). Содержимое сгенерированного
кода скопируйте в отдельную библиотеку - SOAPCLIENT.
    /// Для каждой функции будут созданы классы
Request и Response</remarks> public class
SoapOperation<TIn, TOut>
{

    /// <summary>
    /// Параметры подключения
    /// </summary>
    private SoapConnectionInfo _connectionInfo;
    private string _xmlNamespace;
    /// <summary>
    /// Модуль для которого указывать сообщение в журнале
    /// </summary>
    public ItEventLog.ItModules LogModule =
ItEventLog.ItModules.Adm;
    /// <summary>
    /// Создание экземпляра для вызова функции
    /// </summary>
    /// <param name="info">Параметры подключения. Указание Url
обязательно</param>
    /// <param name="xmlNamespace">Пространство имен запросов и
ответов в конверте Soap</param> public
SoapOperation(SoapConnectionInfo info, string xmlNamespace)
{

```

```

        if (info == null)
            throw new Exception("Не указаны параметры
подключения к веб-сервису"); if
(string.IsNullOrEmpty(info.Url))
            throw new Exception("Не указан
адрес веб-сервиса"); _connectionInfo =
info;
        _xmlNamespace = xmlNamespace;
    }
    /// <summary>
    ///     Выполнить запрос на веб-сервис
    /// </summary>
    /// <param name="obj"></param>
    /// <returns></returns>
    public TOut Execute(TIn obj)
    {
        //     Вызывать на клиенте
        var responseStr =
Operation.DoOnClient<string>("BusinessLogic",
"HttpPostRequest", new object[] {
        _connectionInfo.Url, _connectionInfo.Login ?? string.Empty,
        _connectionInfo.Password
        ?? string.Empty,
getRequestSoapMessage(obj)
});
        //     Получить результат в формате +,<тело ответа> или -
,<сообщение об ошибке>
        //     Если +, то преобразовать результат в необходимый
объект
        //     Если -, записать в журнал событий
        if (!string.IsNullOrEmpty(responseStr))
        {
            var success = responseStr[0] == '+';
            var response = Text.Substring(responseStr, 2);
            if (success)
            {
                InfoManager.MessageBox(response);
                return getResponseFromSoapMessage(response);
            }

            writeToLog(response);
        }

        return default(TOut);
    }

    /// <summary>
    ///     Записать информацию в журнал событий
    /// </summary>
    <param name="message"></param>

```

```

private void writeToLog(string message)
{
    ItEventLog.GenerateEvent(LogModule, _connectionInfo.Url,
message);
}

/// <summary>
/// Преобразовать объект в SOAP-конверт
/// </summary>
/// <param name="obj">Объект для преобразования</param>
/// <returns>Строка представляющая SOAP-
конверт</returns> private string
getRequestSoapMessage(TIn obj)
{
    TypedMessageConverter converter =
TypedMessageConverter.Create(obj.GetType(), null,
_xmlNamespace,
    new XmlSerializerFormatAttribute());
    return converter.ToMessage(obj,
MessageVersion.Soap12).ToString();
}

/// <summary>
/// Преобразовать SOAP-конверт в объект
/// </summary>
/// <param name="soapMessage">Строка представляющая SOAP-
конверт</param>
/// <returns>Объект</returns>
private TOut getResponseFromSoapMessage(string soapMessage)
{
    var xmlReader = XmlReader.Create(new
StringReader(soapMessage));
    var msg = Message.CreateMessage(xmlReader,
Int32.MaxValue, MessageVersion.Soap12);
    TypedMessageConverter converter =
TypedMessageConverter.Create(typeof(TOut), null,
_xmlNamespace, new
XmlSerializerFormatAttribute()); return
(TOut)converter.FromMessage(msg);
}

}

/// <summary>
/// Данные для подключения
/// </summary>
public class SoapConnectionInfo
{
    /// <summary>
    /// URL

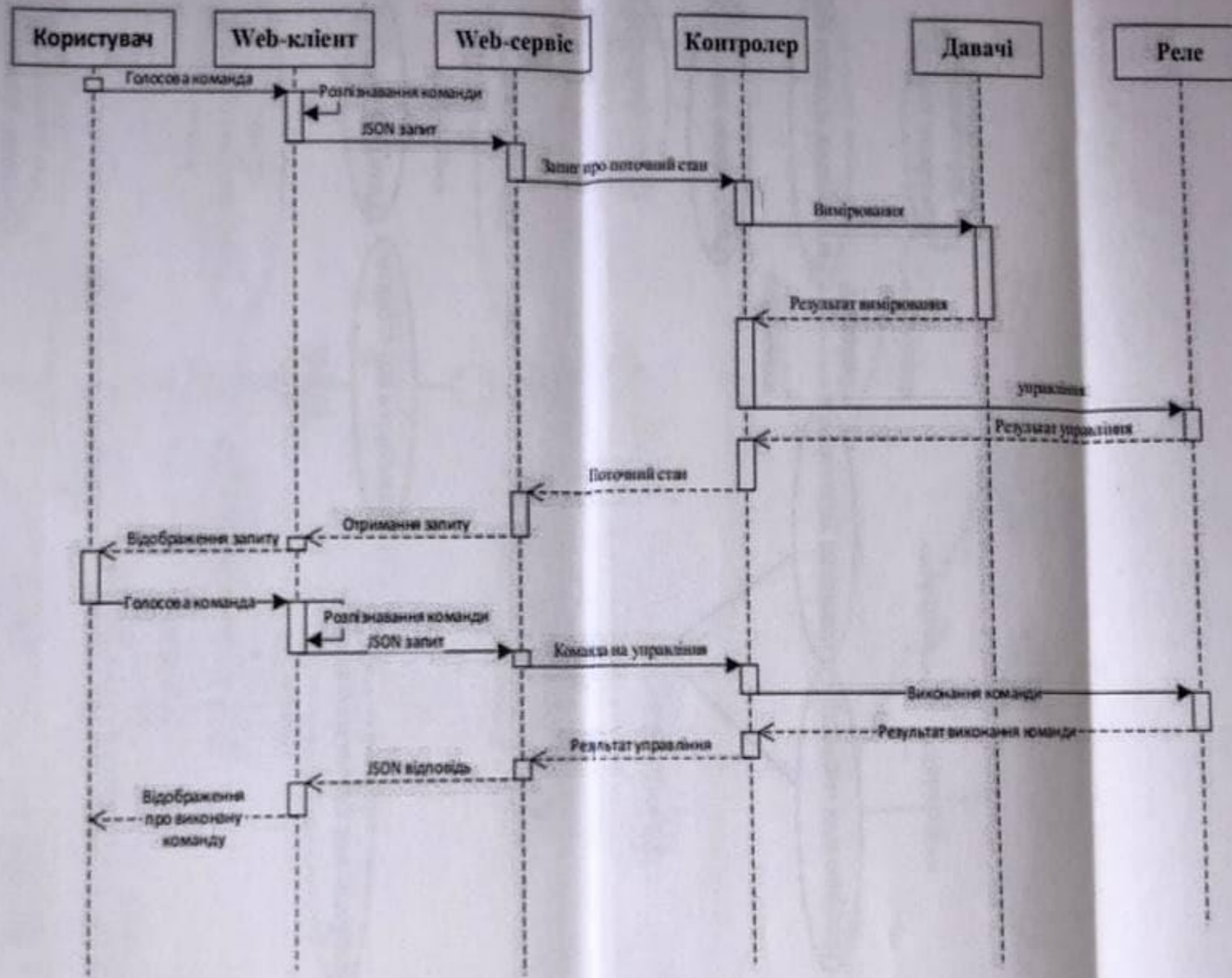
```



```
    /// </summary>
public string Url { get; set; }
    /// <summary>
    /// Логин
    /// </summary>
public string Login { get; set; }
    /// <summary>
    /// Пароль
    /// </summary>
public string Password { get; set; }
}

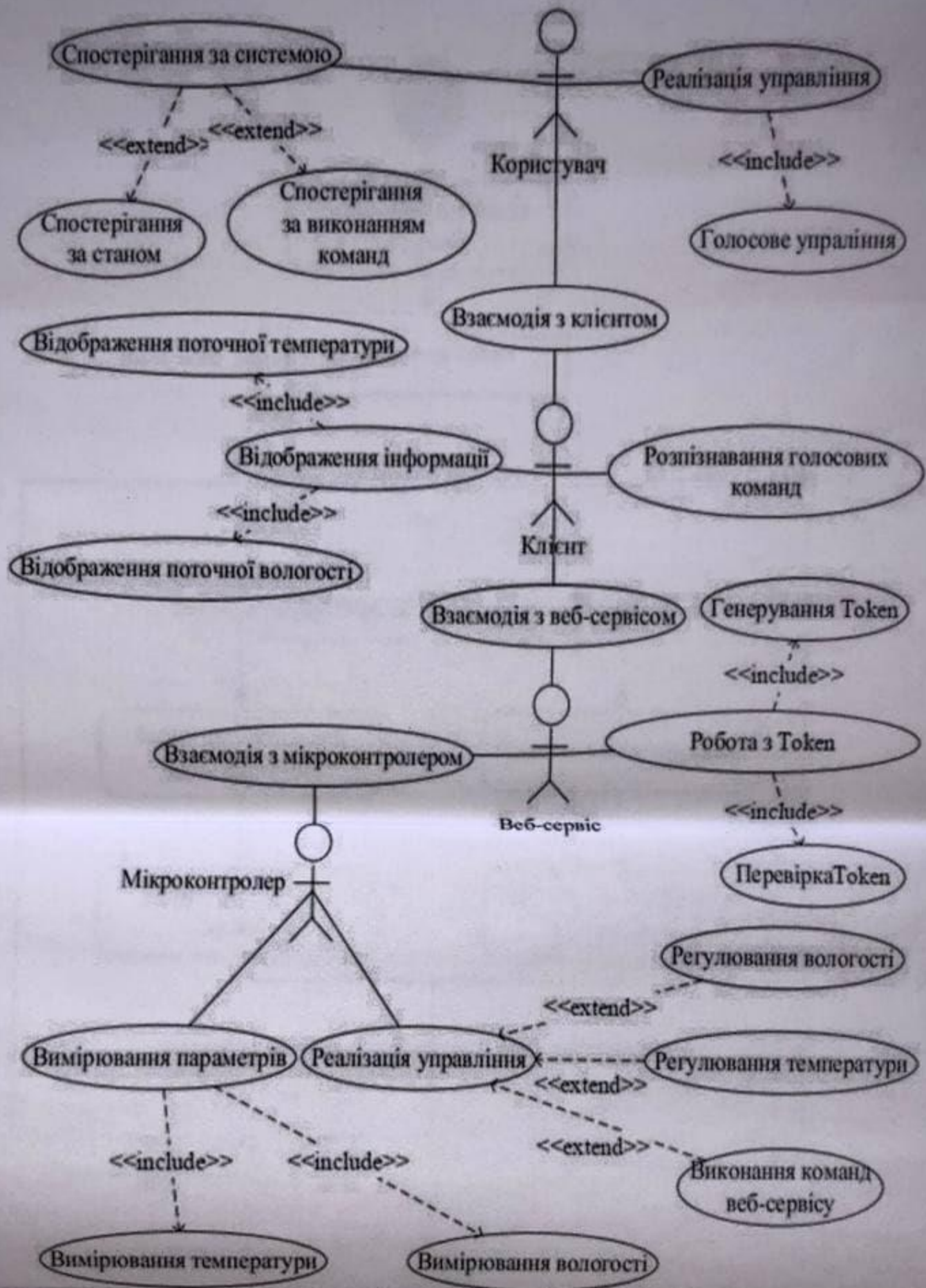
```

# UML-діаграма послідовності



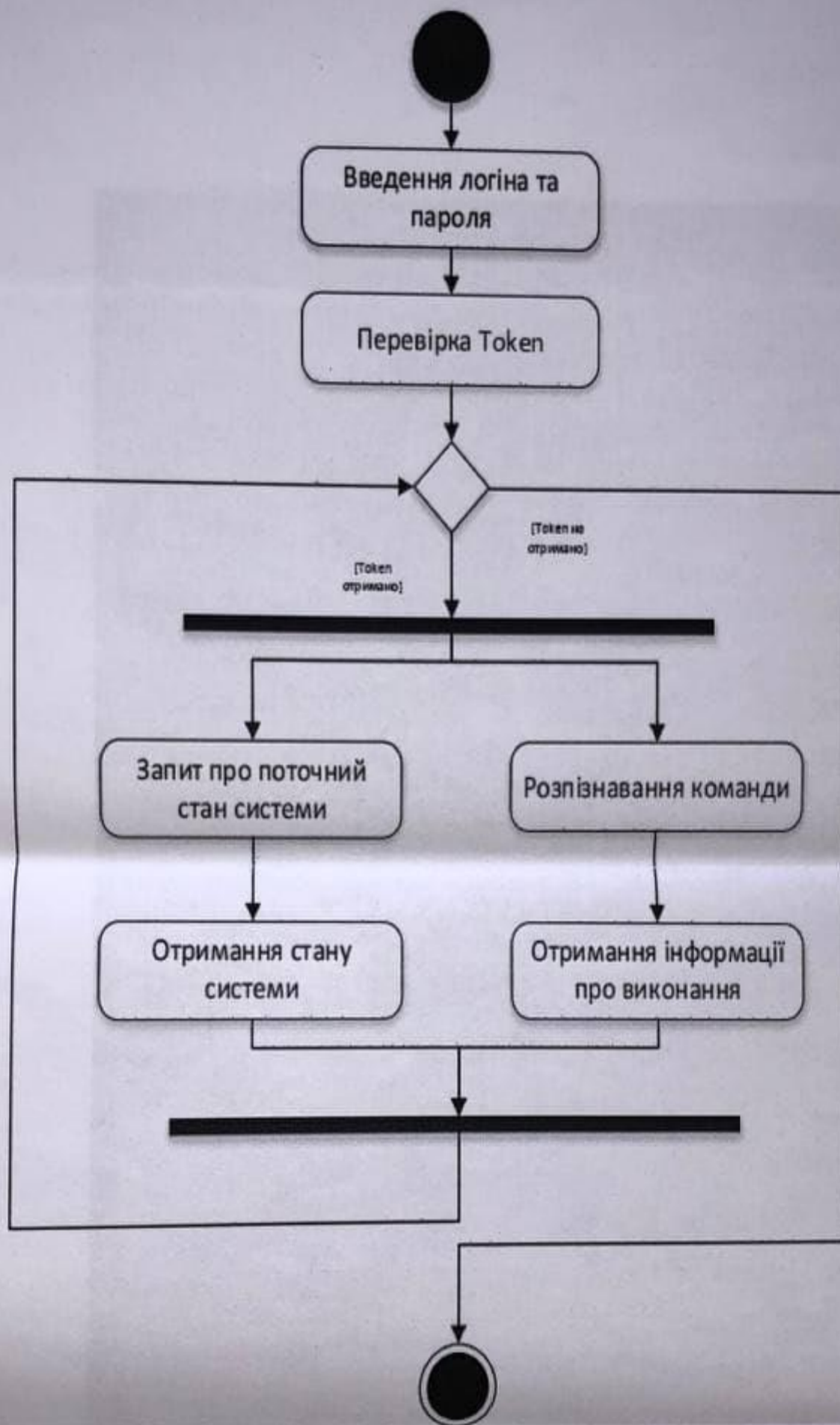
					13.02070849.00022 ПЛ1		
					Розробка системи управління для розумного будинку з використанням веб технологій UML-діаграма послідовності		
Зам.	Лист	№ докум.	Підп.	Дата	Лист	Маса	Масшт.
Розроб.		Шеховцова Т.А.	<i>[Signature]</i>	21.03			
Переа.		Іпьяченко М.Б.	<i>[Signature]</i>	21.03			
Т.контр.					Лист 1		Листів 1
Н.контр.		Щербак Н.В.	<i>[Signature]</i>		НУ «Запорізька політехніка»		
Заст.		Кудерметов Р.К.	<i>[Signature]</i>		КНТ-517		

# UML-діаграма прецедентів



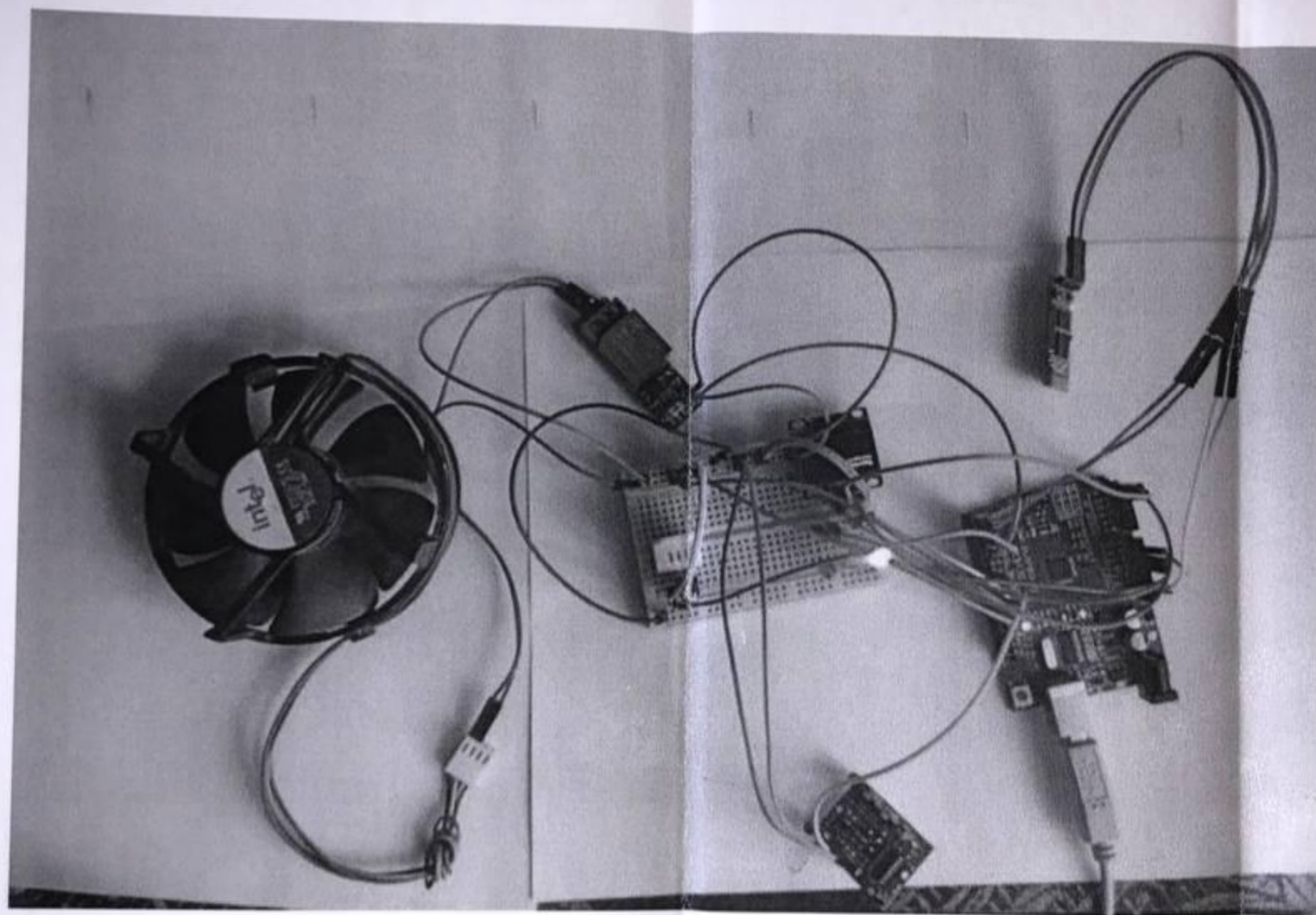
					13.02070849.00022 ПЛ2		
					Розробка системи управління для розумного будинку з використанням веб технологій UML-діаграма прецедентів		
Зам.	Лист	№ докум.	Підп.	Дат.	Лист	Маса	Масшт.
Розроб.		Шеховцова Т.А.	<i>[Signature]</i>	21.05			
Перев.		Ільченко М.Б.	<i>[Signature]</i>	21.05			
Т.контр.					Лист 1		Листів 1
Н.контр.		Щербак Н.В.	<i>[Signature]</i>		НУ «Запорізька політехніка»		
Затв.		Кудерметов Р.К.	<i>[Signature]</i>		КНТ-517		

# UML-діаграма діяльності клієнта



					13.02070849.00022 ПЛЗ		
					Розробка системи управління для розумного будинку з використанням веб технологій		
					UML-діаграма діяльності клієнта		
Зам.	Лист	№ докум.	Підп.	Дата	Лист	Місяц	Місяць
Розроб.		Шеховцова Т.А.	<i>[Signature]</i>	2018			
Перев.		Ільїнко М.Б.	<i>[Signature]</i>	21.0			
Т.контр.					Лист 1		Листів 1
Н.контр.		Щербак Н.В.	<i>[Signature]</i>		НУ «Запорізька політехніка»		
Затв.		Кудерметов Р.К.	<i>[Signature]</i>		КНТ-517		

# Результати роботи



					13.02070849.00022 ПЛ4		
					Розробка системи управління для розумного будинку з використанням веб технологій Результати роботи		
Зам.	Лист	№ докум.	Нідп.	Дата	Лист	Маса	Масшт.
Розроб.		Шеховцова Т.А.	<i>[Signature]</i>	21.08			
Перев.		Іпьяленко М.В.	<i>[Signature]</i>	21.08			
Т.контр.					Лист 1		Листів 1
Н.контр.		Щербак Н.В.	<i>[Signature]</i>		НУ «Запорізька політехніка»		
Затв.		Кудерметов Р.К.	<i>[Signature]</i>		КНТ-517		