

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій та технічного захисту інформацій

Кафедра Радіотехнологій інформаційно-комунікаційних систем

АТЕСТАЦІЙНА РОБОТА Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Інформаційна система збору та обробки даних в сфері комунальних послуг

(тема)

Виконав:

студент II курсу, групи АПСм -22-1

Кириченко А. А.

(прізвище, ініціали)

Спеціальність

126 Інформаційні системи та технології

(код і повна назва спеціальності)

Тип програми

освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма

Архітектурне проектування інформаційних систем

(повна назва освітньої програми)

Керівник

доцент Бітченко О.М.

(посада, прізвище, ініціали)

Допускається до захисту

В.о. зав. кафедри РТКС

(підпис)

Зарудний О.А.

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій і технічного захисту інформації

Кафедра Радіотехнологій інформаційно-комунікаційних систем

Рівень вищої освіти другий (магістерський)

Спеціальність 126 Інформаційні системи та технології

Тип програми Освітньо-професійна

Освітня програма Архітектурне проектування інформаційних систем

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові **КИРИЧЕНКО АЛІНІ АНДРІЇВНІ**

(прізвище, ім'я, по батькові)

1. Тема роботи **ІНФОРМАЦІЙНА СИСТЕМА ЗБОРУ ТА ОБРОБКИ
ДАНИХ В СФЕРІ КОМУНАЛЬНИХ ПОСЛУГ**

затверджена наказом по університету від **3 жовтня 2023 р. № 1295 Ст**

2. Термін подання студентом роботи до екзаменаційної комісії **10 січня 2024 р.**

3. Вихідні дані до роботи _____

3.1 Сформулювати постановку задачі для подальшої розробки

3.2 Розробити структурну схему пристрою збору та обробки даних про витрати електроенергії та води

3.3 Обрати елементну базу для реалізації пристрою

3.4 Розробити користувальницький інтерфейс

4. Перелік питань, що потрібно опрацювати в роботі _____

Вступ. 1 Організація обліку на підприємствах житлово-комунального господарства. 2 Огляд та аналіз аналогічних систем. 3 Схемотехнічна розробка. 4 Програмна розробка. Висновки. Перелік джерел посилання.

Додатки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____

Комп'ютерна презентація

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Вступ	07.10-09.10.2023	
2	Організація обліку на підприємствах житлово-комунального господарства	10.10-25.10.2023	
3	Огляд та аналіз аналогічних систем	26.10-10.11.2023	
4	Схемотехнічна розробка	11.11-03.12.2023	
5	Програмна розробка	04.12-31.12.2023	
6	Висновки	01.01.2024	
7	Оформлення пояснювальної записки	02.01-07.01.2024	
8	Оформлення ілюстрацій	7.01- 09.01.2024	
9	Представлення роботи на кафедрі	10.01.2024	

Дата видачі завдання **3 жовтня 2023 р.**

Студент _____
(підпис)

Кириченко А. А.

Керівник роботи _____
(підпис)

доц. Бітченко О.М.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи магістра містить 82 сторінки тексту, 38 рисунків, 3 таблиць, 15 джерел посилання та 3 додатки.

ДАТЧИК. ІНТЕРФЕЙС. СХЕМА. ЕЛЕКТРОЕНЕРГІЯ. ВОДА.
ІНТЕРНЕТ. МІКРОКОНТРОЛЕР.

Об'єктом розробки є інформаційна система збору та обробки даних в сфері комунальних послуг.

Метод дослідження - описово-аналітичний.

Предметом дослідження є платформа збору та обробки даних з датчиків.

В процесі виконання роботи було проведено огляд та аналіз існуючих систем моніторингу та керування, спроектовано структурну схему пристрою, обрано елементну базу для реалізації пристрою та розроблено користувальницький інтерфейс.

Робота має практичне значення і може бути використана для розробки і впровадження інтегрованих інформаційних систем моніторингу комунальних послуг.

ABSTRACT

Explanatory note of the master's thesis contains 82 pages of text, 38 figures, 3 tables, 15 reference sources and 3 appendices.

SENSOR. INTERFACE. SCHEME. ELECTRICITY. WATER.
INTERNET. MICROCONTROLLER.

The object of development is an information system of data collection and processing in the field of communal services.

The research method is descriptive and analytical.

The subject of research is a platform for collecting and processing data from sensors.

In the course of the work, a review and analysis of the existing monitoring and control systems was carried out, a structural diagram of the device was designed, an element base was selected for the implementation of the device, and a user interface was developed.

The work is of practical importance and can be used for the development and implementation of integrated information systems for monitoring utility services.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів.....	6
Вступ.....	8
1 Організація обліку на підприємствах житлово-комунального господарства	10
1.1 Принципи побудови комерційного обліку.....	10
1.2 Деякі аспекти впровадження автоматизованих систем.....	14
2 Огляд та аналіз аналогічних систем.....	20
2.1 Принципи побудови інформаційних систем.....	20
2.2 Автоматизована система Tera Bills.....	20
2.3 Автоматизована система управління підприємствами житлово- комунального господарства (АСУП ЖКГ).....	22
2.4 Одеська автоматизована система прийому та обліку комунальних платежів (АСУ КП) ГЕРЦ.....	24
2.5 Комплексна автоматизована система «ЕЛЛІС-ЖКГ».....	25
2.6 Система ISpro.....	26
2.7 Бездротова мережева технологія LoRa.....	29
2.8 Значення сучасних IoT в сьогодні.....	32
3 Схемотехнічна розробка.....	34
3.1 Розробка структурної схеми пристрою.....	34
3.2 Вибір датчика витрати електроенергії.....	34
3.3 Вибір датчика витрат води.....	38
3.4 Вибір мікроконтролера.....	42
4 Програмна розробка.....	44
4.1 Розробка програми роботи мікроконтролера.....	44
4.2 Розробка Web додатку.....	46
Висновки.....	58
Перелік джерел посилання.....	59
Додаток А – Програмні коди.....	61
Додаток Б – Слайди презентації.....	69
Додаток В – Відомість кваліфікаційної роботи.....	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AWS (Amazon Web Services) – хмарна платформа, за допомогою якої можна обмінюватися даними незалежно від місцезнаходження;

GSM (Global System for Mobile Communications) – глобальна система мобільного зв'язку міжнародний стандарт для мобільного цифрового стільникового зв'язку;

IoT (Internet of Things) – концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані давачі, а також програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі;

LoRa (Long Range) – це новий бездротовий протокол, розроблений саме для зв'язку на великі відстані та зв'язку з низьким енергоспоживанням;

LoRaWAN (Long Range Wide Area Network) – це малопотужна бездротова двоспрямована мережева технологія для передачі даних малого обсягу, яка охоплює великі відстані;

PLC (Programmable logic controller) – «програмований логічний контролер», комп'ютерний пристрій, що використовується промисловими системами управління;

Wi-Fi (wireless fidelity) – «бездротова надійність» застосовується при з'єднанні великої кількості комп'ютерів, бездротове підключення до мережі;

АЦП – аналого-цифровий перетворювач;

БД (database) – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами;

БКБ – багатоквартирні будинки;

ЖКГ – житлово комунальне господарство;

ІС – інформаційна система;

ІТ – інформаційні технології;

ПЗ – програмне забезпечення;

ПЗПД – пристрій збору та передачі даних;

ТВЖ – товариство власників житла;

ЦАП – цифро-аналоговий перетворювач;

ЦОД – центр обробки даних.

ВСТУП

В наш час врахування споживання електроенергії, гарячої та холодної води, теплоенергії в ЖКГ (житлово-комунальне підприємство) останнім часом стає актуальним завданням для всієї країни. Сьогодні на ринку подібних систем представлені лише спеціалізовані системи, такі як системи автоматизованого обліку електроенергії. Починають з'являтися перші моделі систем, які здійснюють облік як електроенергії, та й водоспоживання.

Облік теплоенергії, що передається в житло споживачів на сьогоднішній момент відбувається не за кількістю спожитої енергії, а за метражем житлової площі (у деяких випадках встановлюються загальнобудинкові теплові лічильники, які розраховують кількість спожитого тепла всім будинком і сплат за спожиту теплоенергію розподіляється пропорційно площі квартири щодо загальнобудинкової площі). Створення системи комплексного обліку енерго- та водопостачання дозволяє вирішити досить великий обсяг завдань, пов'язаних не лише з урахуванням, але й формування тарифних ставок. Постачальники отримують у максимально можливому обсязі гроші за надані ресурси, а споживачі, своєю чергою, платять лише за те, що споживали. Поруч із існує поняття комерційних втрат під час постачання комунальних ресурсів споживачам, у яких постачальники змушені компенсувати ці втрати з допомогою споживачів, тому розробка інформаційної системи з комплексному обліку споживання енерго- та водо- ресурсів є актуальним завданням.

Таким чином, на сьогоднішній день проблема побудови інформаційних систем з комплексного обліку енергопостачання в ЖКГ залишається одним з актуальних завдань, незважаючи на те, що існує досить велика кількість публікацій на цю тему, інформаційні системи з обліку комунальних ресурсів є розрізненими проектами та актуальним завданням стає створення методичного та програмного забезпечення для розробки інформаційних систем.

Метою роботи є розробка автоматичної системи збору даних комунальних послуг з розрахунком вартості за місяць.

Задачі роботи:

- провести огляд аналогічних систем;
- розробити структурну схему пристрою збору даних;
- обрати елементну базу для реалізації пристрою;
- розробити відповідне програмне забезпечення.

1 ОРГАНІЗАЦІЯ ОБЛІКУ НА ПІДПРИЄМСТВАХ ЖИТЛОВО-КОМУНАЛЬНОГО ГОСПОДАРСТВА

1.1 Принципи побудови комерційного обліку

У процесі дослідження, нами детально розглянуто принципи побудови комерційного обліку з надання комунальних послуг. Метою такого виду обліку – є встановлення не лише якісних, а й кількісних характеристик наданої послуги. Якраз встановлення ефективної системи облікового відображення згаданих послуг, сприятиме удосконаленню розрахунків за комунальні послуги.

Суб'єктами правовідносин при цьому є: оператори інженерних мереж, виконавці послуг, виконавці розподілу, власники (співвласники) будинку. Визначення законодавчо закріплених суб'єктів у процесі надання послуг з водозабезпечення та водовідведення є механізмом організації ефективної системи бухгалтерського обліку на досліджуваних підприємствах (таблиця 1.1) [1].

Таблиця 1.1 – Суб'єкти правовідносин комерційного обліку

Суб'єкти правовідносин	Функції суб'єктів
Оператори зовнішніх інженерних мереж	Інформує про намір встановити та оснащує вузлами комерційного обліку; обслуговує ВКО; приймає на абонентський облік ВКО
виконавці послуг	приймає на абонентський облік ВКО; знімає показання ВКО; збирає внески за оснащення ВКО; приймає на абонентський облік ВРО; функціональна перевірка ВРО
виконавці розподілу	знімає показання ВКО; збирає внески за оснащення ВКО; приймає на абонентський облік ВРО; функціональна перевірка ВРО
власники (співвласники) будинку	дає згоду/ самостійно оснащує будівлю ВКО
споживач	сплачує внески за встановлення, заміну та обслуговування ВКО, зняття показань ВРО

У процесі дослідження нами з'ясовано проблеми в розвитку житлово-комунальної сфери. Відповідно, основними шляхами є удосконалення наступних складових економічної політики підприємства: цінової та тарифної політики; підвищення якості послуг; створення інституту відповідальних

власників; політики соціального захисту; фінансово-стабілізаційної політики; системи управління; нормативно-законодавчої бази тощо.

Регулювання взаємовідносин у сфері ЖКГ здійснюється на національному та регіональному рівні. Важливо знати, що галузь дослідження регулює низка законопроектів основні із яких наведені на рисунку 1.1 [2].

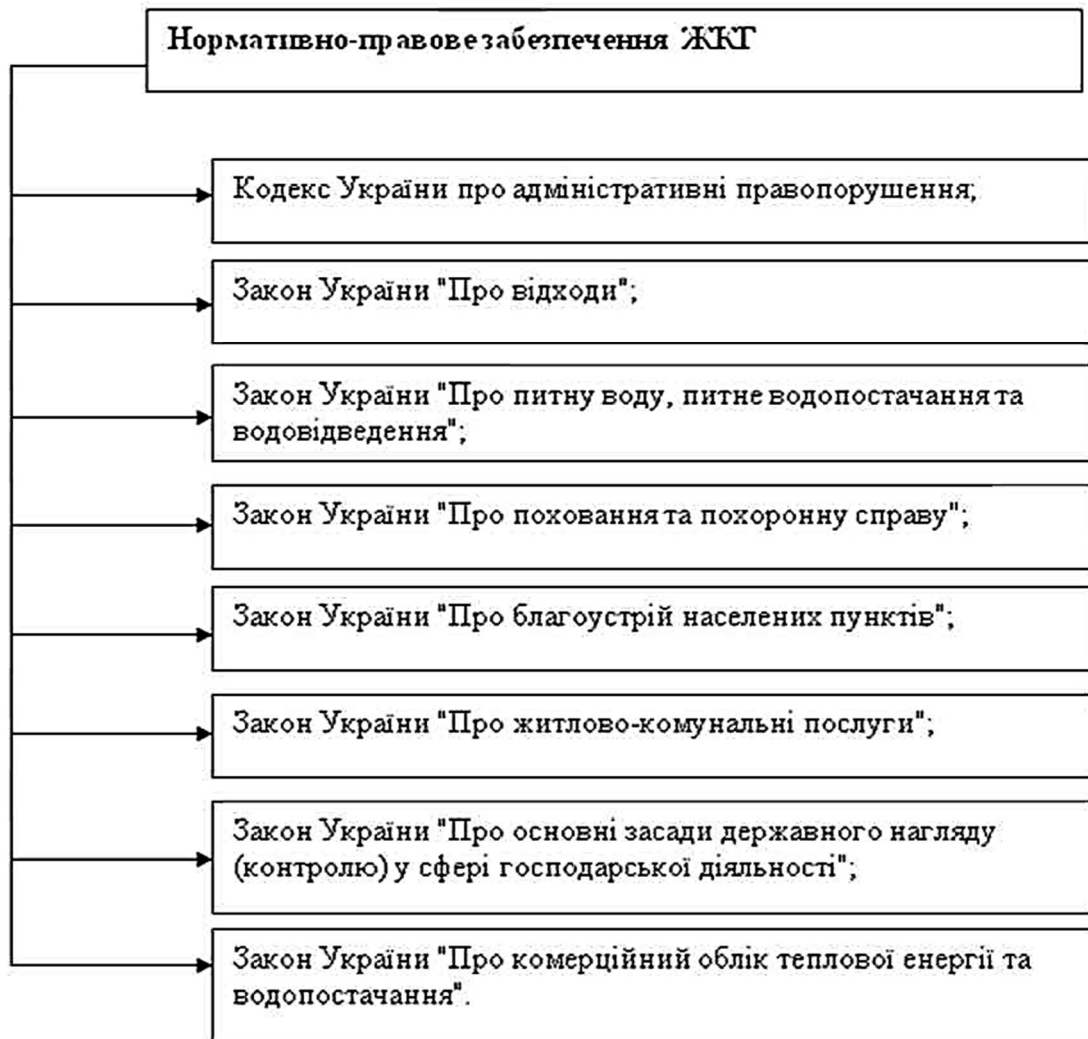


Рисунок 1.1 – Нормативно-правове забезпечення ЖКГ

Основні особливості, які мають безпосередній вплив на організацію обліку підприємств ЖКГ деталізовано у таблиці 1.2.

Таблиця 1.2 – Особливості ЖКГ

Характерні особливості	Принципи організації
основні особливості	висока соціальна значимість галузі, одночасність виробництва і споживання ЖКП та відсутність запасів готової продукції, місцевий характер виробництва і споживання послуг, великі площі обслуговування, різноманітність видів робіт.

Продовження таблиці 1.2

Характерні особливості	Принципи організації
похідні особливості	<p>оцінку ефективності діяльності підприємств ЖКГ здійснюють за соціально економічними показниками (а не тільки економічними), держава здійснює контроль діяльності підприємств галузі і якості ЖКП, одним із основних напрямів якого є державне регулювання тарифів (воно здійснюється шляхом встановлення граничних рівнів рентабельності, обмеження росту витрат (зокрема, заробітної плати) підприємств-монополістів, встановлення рівня відшкодування витрат на житлово-комунальні послуги, затвердження розміру тарифів тощо).</p> <p>Крім того, в ЖКГ специфічна структура основних засобів і затрат: у структурі основних засобів ЖКГ переважають передавальні пристрої (теплові, водопровідні, каналізаційні мережі), багато невиробничих основних засобів (насамперед житлові будинки), що вимагає великих капітальних затрат на створення та утримання основних засобів; у структурі запасів незначна питома вага витрат на матеріали при значній енергоємності галузі та досить суттєвому рівні витрат на амортизацію.</p>
особливості, пов'язані з сучасними умовами	<p>зполягають у недостатньому рівні якості і кількості житлово-комунальних послуг, що обумовлено, зокрема, технічною відсталістю галузі, нераціональним використанням ресурсів як підприємствами ЖКГ, так і споживачами, невідповідністю житла вимогам та невизначеністю питань власності на нього. Своєю чергою, технічна відсталість галузі обумовлена недостатністю тарифів на ЖКП та фінансування капітальних інвестицій на поліпшення ОЗ при значній заборгованості населення і бюджету, що призводить до збитковості підприємств ЖКГ.</p>

Основною особливістю, яка безпосередньо впливає на підгалузь водопостачання та водовідведення – монопольний характер у процесі надання таких послуг. Тобто впливати на зменшення вартості наданих послуг споживачами є практично неможливим.

Основними факторами впливу особливостей економіки ЖКК на облік є такі: методика ціноутворення, різноплановість видів робіт, значна кількість споживачів послуг, значна кількість споживачів, недостатність тарифів і фінансування (таблиця 1.3).

Таблиця 1.3 – Фактори впливу на побудову системи обліку

Фактори впливу	Особливості впливу
діючий порядок ціноутворення	зумовлює отримання цільового фінансування (субсидій малозабезпеченим громадянам, пільг, дотацій тощо) та необхідність визначення не тільки виробничої, а й повної собівартості послуг;
різний порядок управління житлом при різних власниках	обумовлює різний порядок ведення обліку (в житловій організації, в управителя житла або в об'єднаннях співвласників багатоквартирних будинків
стохастичність процесів виробництва, різноманітність видів робіт	ускладнюють нормування та облік виконаних робіт і наданих послуг;
особливості складу та структури активів і витрат	визначають специфіку їх обліку;
велика кількість споживачів послуг і платежів кожного з них	обумовлює складність обліку розрахункових операцій;
послуги надаються відомому споживачу, але при цьому попередньо невідомому обсязі,	робить практично неможливим визначення розподілених і нерозподілених загальновиробничих витрат;
недостатність тарифів і фінансування	обумовлює збитковість підприємств галузі.

Підприємства ЖКК є комунальними підприємствами, які функціонують у вітчизняній податковій системі як і всі інші суб'єкти господарювання, сплачуючи відповідні податки, проте, зважаючи на певні особливості галузі, їм надається ряд пільг в оподаткуванні. Зокрема, до місцевих бюджетів зараховуються: комунальний податок та частина плати за землю. При цьому органи самоврядування можуть даним суб'єктам надавати пільги з оподаткування.

Отже, особливості функціонування галузі визначають особливості обліку (рисунок 1.2).



Рисунок 1.2 – Облік на підприємствах ЖКГ

1.2 Деякі аспекти впровадження автоматизованих систем

На сьогоднішній день комунальні платежі є самими масовими платежами на всій території країни, причому технологія нарахування, прийому й обробки платежів істотно не помінялася. У більшості населених пунктів

нарахуваннями комунальних платежів займаються самі виробники комунальних послуг, а прийом платежів здійснюють банківські установи. У виді явних зловживань з боку ЖЕК (при перерахуванні коштів) багато комунальних служб відмовилися від їх послуг по нарахуванню і зборів засобів за зроблені послуги тепло, гарячу воду, холодну воду, газ, електроенергію і кинулися самі проводити нарахування і збирати кошти безпосередньо з населення.

На багатьох комунальних підприємствах були створені свої абонвідділи, укладені договори з банківськими установами, поширені свої розрахункові книжки серед населення. У такий спосіб підприємства комунального господарства вирішили проблеми надходження засобів безпосередньо на свої розрахункові рахунки, але, на жаль, з'явилися нові проблеми. У виді того, що нарахування виробляються в основному по нормі, а норма нараховується на душу населення, кожному з комунальних підприємств довелося вести облік по кожному абоненті з урахуванням прописки-виписки громадян. До цієї проблеми додалися ще дві – пільги і субсидії. Тому що джерелом усіх змін в абонента, що істотно впливають на нарахування, є паспортний стіл, розташований у ЖЕК, то в комунальних служб почали накопичуватися помилки в нарахуваннях. Такі помилки, як правило, не виявляються швидко і на виправлення вимагають багато зусиль. У теж час, банки не тільки по колишньому затримують перерахування зібраних засобів, а і намагаються підвищити відсоток комісійної винагороди, пояснюючи це тим, що зросли обсяги платежів. А абонентів приходиться з'являтися хоча б раз у рік у кожній комунальній службі для перерахунку й уточнення боргів із усіма платіжними квитанціями. Для аналізу ситуації все різноманіття при прийомі комунальних платежів сьогодні можна звести до трьох основних методів [3]:

- кожен сам за себе;
- розрахункові центри при банках;
- незалежний розрахунковий центр.

При методі кожен сам за себе всі комунальні господарства проводять нарахування самі, силами своїх абонентських відділів (рисунок 1.3).

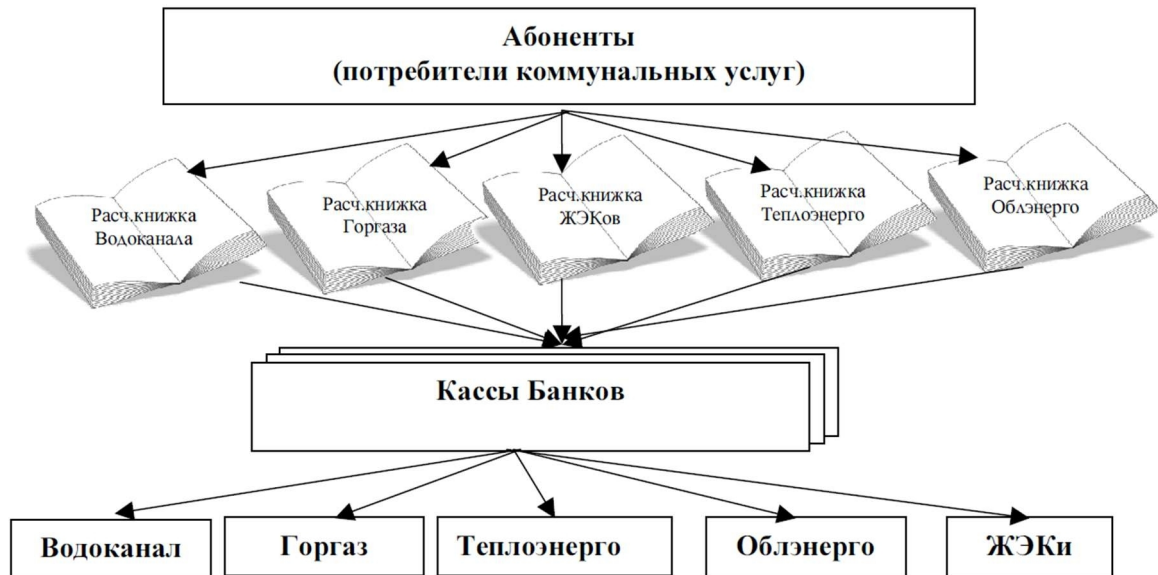


Рисунок 1.3 – Метод розрахунків кожен сам за себе

Комунальні платежі приймають також на свої розрахункові рахунки, уклавши договори з банківськими установами або відкривши свої власні каси. Способи прийому грошей від населення розрізняються: деякі друкують рахунки і розносять їхнім абонентам, деякі просто надрукували і роздали абонентам розрахункові книжки, а деякі домовилися з банками і передають їм дані про борги абонента в “машинній формі”. А банки за допомогою свого програмного забезпечення проводять прийом комунальних платежів по кожній комунальній службі окремо і перелічують засобу безпосередньо на комунальну службу.

Основні переваги цього методу:

- посередники між комунальними господарствами і населенням зведені до мінімуму (тільки банк або прийом ведеться через власні каси);
- за нарахування по комунальних платежах відповідає безпосередньо виробник комунальних послуг;
- можливість передавати усі в “машинній формі” спрощує обробку платежів у випадку з передачею інформації про борги безпосередньо банкам.

Основні недоліки:

- незручність населенню (багато книжок або рахунків);

- велика кількість помилок, тому що ведуть базу даних по абонентах на кожному підприємстві;
- відсутній контроль за прийомом коштів і перерахуваннями банку;
- невиправдано високі витрати (базу даних необхідно вести на кожному підприємстві);
- багато проблем при наданні пільг і субсидій. Для того щоб призначити субсидію, усі комунальні підприємства повинні дати свої дані по нарахуваннях;
- у кожній комунальній службі необхідно вводити інформацію про оплату.

При використанні розрахункових центрів при банках один з великих банків виступає лідером при прийомі комунальних платежів (рисунок 1.4). Він розробляє правила і примушує інші банки грати по них.

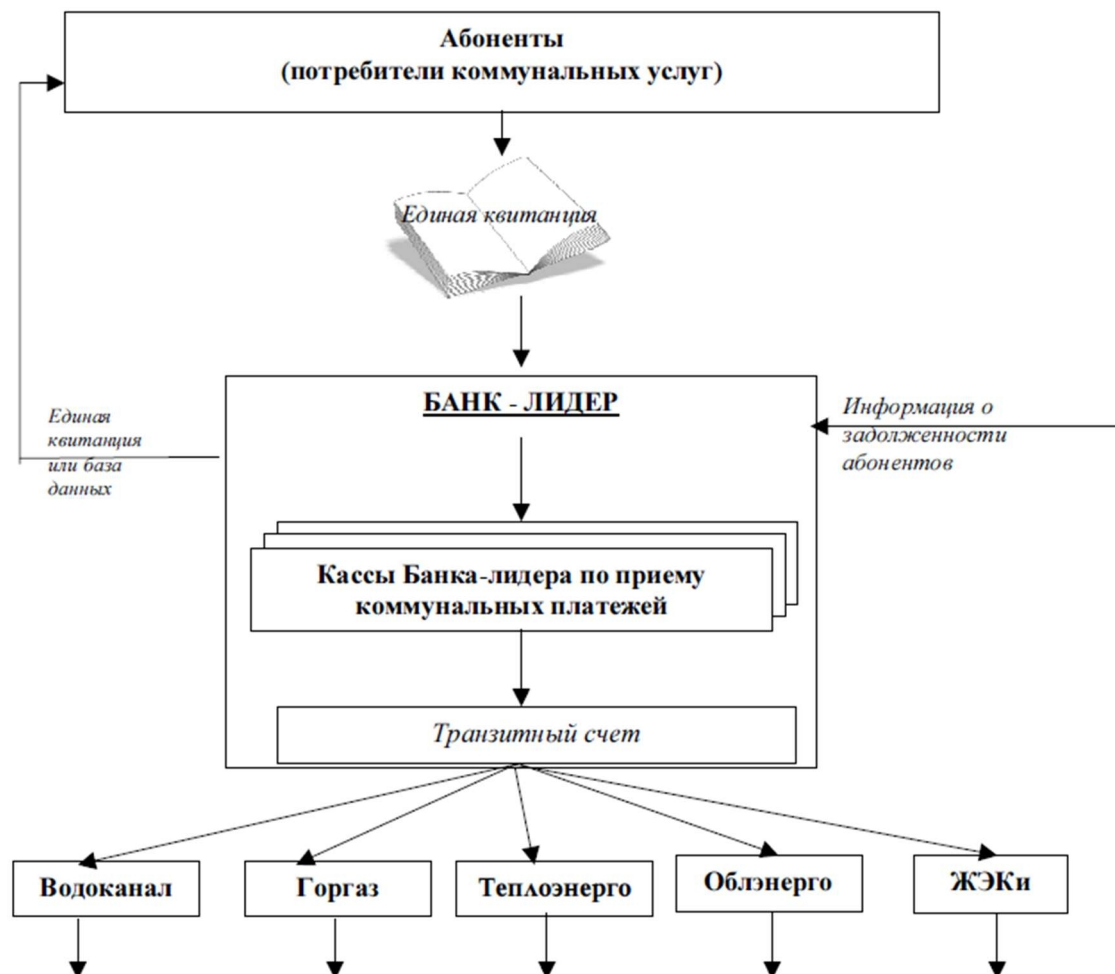


Рисунок 1.4 – Метод використання розрахункових центрів при банках

Розрахунок нарахувань може вироблятися як у розрахунковому центрі банку, так і в абонвідділі кожного комунального господарства, але всі дані про нарахування надходять у розрахунковий центр у “цифровому” виді. Банк може використовувати як спеціальне програмне забезпечення, так і єдині розрахункові книжки (для всіх послуг). Вся оплата за комунальні платежі збирається на транзитному рахунку в банку-лідері. Через технологічний час банк перелічує відповідну частину засобів на рахунки комунальних служб.

Основні переваги:

- зручно населенню (одна розрахункова книжка або рахунок);
- можливість “машинної” обробки платежу;
- передумови створення єдиної бази абонентів;
- можливість передачі централізованих даних у відділи субсидій.

Основні недоліки:

- збір і розподіл засобів із транзитного рахунка значно подовжує одержання засобів комунальними підприємствами.
- явне лідерство одного з банків створює проблеми іншим банкам-конкурентам (банк-лідер може і не передавати зведення по боргах населення іншим банкам, істотно знижувати комісійну винагороду іншим банкам або взагалі не укладати з ними договору тощо);
- банк-лідер зацікавлений як можна довше зберігати засобу населення на своєму транзитному рахунку (у якості безкоштовного кредитного ресурсу).
- тому що розрахунковий центр знаходиться в банку, а банк зацікавлений тільки безпосередньо в банківських операціях, то очікувати від розрахункового центра плідної роботи з комунальними підприємствами і з відділами пільг і субсидій не приходиться. Мова насамперед йде про програми перегляду і контролю нарахування, роботи з боржниками, консультаціями населення, видачі довідок тощо.

При незалежному розрахунковому центрі всі нарахування робить єдиний розрахунковий центр, ведучи при цьому міську базу даних по всіх абонентах комунальних платежів у розрізі всіх послуг. Раз на місяць розрахунковий центр друкує повідомлення і квитанції по кожному абоненті. Розрахунковий

центр укладає договори з банківськими установами на прийом платежів і сам розносить платежі по кожнім абоненті, іншими словами, веде весь облік по абонентах. Грошові кошти розрахунковий центр, як правило, збирає на єдиний транзитний рахунок, і через визначений час проводить перерахування комунальним службам (рисунок 1.5).

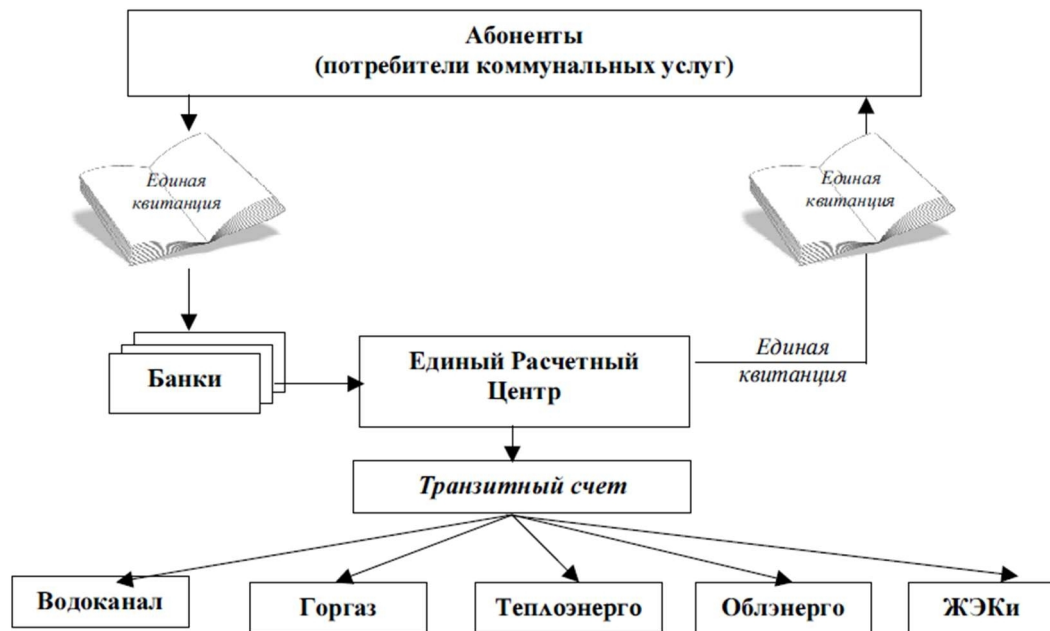


Рисунок 1.5 – Незалежний розрахунковий центр

Основні переваги:

- зручно населенню (єдиний рахунок на всі послуги);
- мінімальна кількість помилок;
- мінімальні втрати комунальних служб на виробництво нарахування і прийом платежів;
- створення здорової конкуренції між банками;
- можливість побудови на основі єдиної міської бази абонентів різних довідкових служб.

Недоліки:

- використання транзитного рахунка сповільнює одержання засобів комунальними підприємствами.
- надруковані єдині повідомлення і квитанції позбавляють можливість населення платити частково, або не оплачувати деякі послуги.

2 ОГЛЯД ТА АНАЛІЗ АНАЛОГІЧНИХ СИСТЕМ

2.1 Принципи побудови інформаційних систем

Завдяки розвитку комунікацій та обчислювальної техніки побудова такої системи є нескладним завданням. При побудові системи застосовуються такі принципи [4]:

- самостійне ведення нарахувань з усіх видів платежів;
- перегляд необхідної інформації про нарахування та оплату комунальних платежів у «реальному часі»;
- всі учасники системи підключаються до єдиної бази абонентів (кожен зі своїми повноваженнями, визначеними у правилах) у “реальному часі”;
- можливість формувати та зберігати “електронний рахунок” за кожною комунальною послугою за всі періоди нарахування;
- перерахування коштів банківськими установами має здійснюватися безпосередньо на розрахункові рахунки відповідних комунальних служб;
- сувора персональна відповідальність за будь-які внесені дані до системи;
- надавати в “реальному часі” “електронні рахунки” з відображенням минулої оплати та коригування за будь-яким абонентом, згруповані за періодом нарахування, у всі можливі місця оплати та перегляду.

2.2 Автоматизована система Tera Bills

Система Tera Bills - це програмний комплекс, що виконує розрахунок нарахувань з оплати житлово-комунальних послуг, облік, накопичення та обробку інформації про надходження платежів квартиронаймачів, документування та відображення даних, а також формування

різноманітних звітів. Основні функції та можливості Tera Bills:

- ведення баз даних по житловому фонду, мешканцям, особовим рахункам;
- розрахунок нарахувань по кожному особовому рахунку з урахуванням поточних тарифів та наявності пільг по всьому обсягу послуг з можливістю перерахунку;
- розрахунок нарахувань за комунальні послуги у разі наявності індивідуальних чи групових лічильників, а також підтримка кількох лічильників для одного особового рахунку;
- перерахунок нарахувань у разі зміни ставок та тарифів на послуги, що надаються населенню;
- можливість друку платіжних квитанцій на лазерному (струменевому) принтері. Друк інформації про особовий рахунок з використанням шаблонів, що налаштовуються;
- автоматизація прийому платежів;
- формування стандартизованих звітів у MS Excel;
- формування звітів на запит користувача (на основі шаблонів, що налаштовуються) (рисунок 2.1) [5].

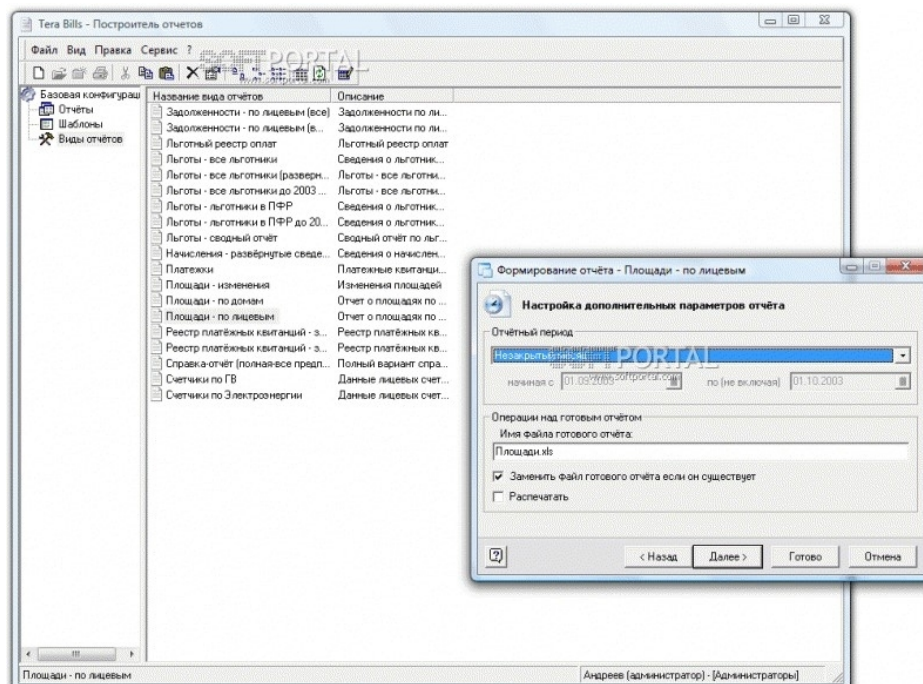


Рисунок 2.1 – Интерфейс программы Tera Bills «Формування звіту»

2.3 Автоматизована система управління підприємствами житлово-комунального господарства (АСУП ЖКГ)

АСУП ЖКГ дає принципово нові можливості управління - від отримання необхідної інформації до прийняття та контролю за результатами управлінських рішень.

Переваги АСУП ЖКГ:

- наявність оперативної інформації щодо кожного об'єкта ЖКГ – стан, заплановані та проведені роботи;
- планування та контроль платежів, бюджетів на утримання та ремонт об'єктів ЖКГ;
- контроль необхідних показників у режимі on-line;
- підвищення якості робіт та зниження аварійності;
- єдиний розрахунковий центр із прийому платежів, формування рахунків за спожиті послуги, контроль оплат, перерахунки за недоотримані послуги;
- потужний аналітичний апарат, можливість моделювання ситуації «що-якщо»;
- ефективне та «прозоре» планування роботи підприємств ЖКГ;
- підвищення рентабельності підприємств ЖКГ.

Підсистема "Облік послуг з водопостачання, водовідведення та інших комунальних послуг" (конфігурація "Водоканал, ЖЕК, ОСББ") програми "ДЕБЕТ Плюс" призначена для організації повного циклу обліку наданих послуг та взаєморозрахунків із споживачами у комунальній сфері. Довідник тарифів на послуги ведеться у розрізі зручностей та категорій споживачів. У ньому зберігається інформація про норми споживання для можливості визначення тарифів для об'єктів без лічильників. У системі передбачено довідник пільг, у якому кожної пільгової категорії населення вказується відсоток пільги кожному окремо взяту послугу. Центральним компонентом підсистеми є довідник особових рахунків, у якому зберігається вся інформація про види зручностей,

наявність пільг та субсидій (рисунок 2.2) [6].

Справочник аналитики: редактирование

Лицевой счет: 8 ФИО(Название): Рудюк З.А.

Тип абонента: физическое лицо Категор. потреб.: 1 Населення

открыт: 01.11.2006 закрыт: .19

Насел. пункт: 1 смт Білогір'я

Улица: 22 Шевченка

№ дома: 40 Квартира: 5 Корпус:

Реквизиты физического лица: Идент. код:

Паспорт серия: номер: дата: .19

место выдачи:

Реквизиты организации: ЄДРПОУ:

Является плат. НДС: Г № свидет.: Код НДС:

Удобства

Период		Удобства		Льготы		Количество	
с	по	код	название	код	название	по счетчику	без счетчика
01.11.2006		2	Водоснабжение	0		-	1,0
01.11.2006		2	Водоснабжение	2	Учасники ВВВ	-	1,0
01.11.2006		2	Водоснабжение	4	діти війни	-	2,0

Субсидии

Период		Услуга		Сумма
с	по	Код	Название	
01.03.2007		99.01	Водоснабжение	20,0

Сохранить Параметры Расч.счета Расширения Отмена

Рисунок 2.2 – Довідник особових рахунків

Система зберігає всю історію змін тарифів, норм споживання, наявності пільг та субсидій, що дає унікальну можливість автоматичного проведення перерахунків за попередні періоди у разі зміни тарифів „заднім числом” або при виявленні помилок. Засобами системи формується звітність щодо сум наданих пільг та субсидій. Для цих цілей у системі передбачено довідник пільговиків, у якому перераховуються всі особи, які користуються пільгами із зазначенням видів та реквізитів документів, що дають право на пільги. У системі передбачено зручний режим занесення даних за показаннями лічильників, коли початкові показання автоматично беруться з інформації за попередній місяць. Для запобігання виникненню помилок при коригуванні даних за попередні періоди передбачено відповідний контроль та автоматизоване виправлення даних наступних періодів. На підставі нарахувань за послуги система формує споживачам рахунки на оплату послуг із зазначенням сум заборгованості та розшифровкою показань лічильників. Зовнішній вигляд рахунків узгоджується на етапі застосування системи (рисунок 2.3).

Платіжне пов'язування до рахунку № 8058 від 01.11.2006 О.Р. 904 листопад 2006 року Отримувач: КП "Білогір'яводоканал" Платник: Мацюк І. Сума за надані послуги 38,31 грн.	Комунальне підприємство "Білогір'я водоканал" Рахунок № 8058 від 01.11.2006 на оплату за надані послуги за листопад 2006 року Споживач: Мацюк І. Адреса: Отримувач: КП "Білогір'яводоканал" № р/р 26005052405259 в банку ХФ КБ "Приватбанк"					Зберігати 3 роки Особовий рахунок 904																			
	<table border="1"> <tr> <td>Сума боргу на 01.11.2006 грн.</td> <td>Нараховано всього грн.</td> <td>Відшкодування пільги грн.</td> <td>Оплата грн.</td> <td>Субсидія грн.</td> <td>Сума боргу на 01.12.2006 грн.</td> </tr> <tr> <td>0</td> <td>38,31</td> <td>0</td> <td>38,31</td> <td>0</td> <td>38,31</td> </tr> </table>	Сума боргу на 01.11.2006 грн.	Нараховано всього грн.	Відшкодування пільги грн.	Оплата грн.	Субсидія грн.	Сума боргу на 01.12.2006 грн.	0	38,31	0	38,31	0	38,31	Показники лічильників: <table border="1"> <tr> <td>№ ліч.</td> <td>попер.</td> <td>кінцеві</td> <td>використ.</td> </tr> <tr> <td>658987</td> <td>4893</td> <td>4973</td> <td>80</td> </tr> </table>					№ ліч.	попер.	кінцеві	використ.	658987	4893	4973
Сума боргу на 01.11.2006 грн.	Нараховано всього грн.	Відшкодування пільги грн.	Оплата грн.	Субсидія грн.	Сума боргу на 01.12.2006 грн.																				
0	38,31	0	38,31	0	38,31																				
№ ліч.	попер.	кінцеві	використ.																						
658987	4893	4973	80																						
Просимо заповнити Покази лічильника <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> Дата _____.20__р. Підпис абонента _____	До оплати: 38 грн. 31 коп. Термін оплати на протязі 10 днів після одержання рахунку Рахунок дійсний до: 25 грудня 2006 При потребі звертатись в КП "Білогір'яводоканал" за тел. (03841) 9-23-95, 9-15-74, смт. Білогір'я, в ул. І. Франка 60 Суми і тариф вказані з врахування ПДВ																								

Рисунок 2.3 – Зовнішній вигляд рахунків

2.4 Одеська автоматизована система прийому та обліку комунальних платежів (АСУ КП) ГЕРЦ

ГЕРЦ створена з метою упорядкування розрахунків за енергоносії та забезпечення ефективного контролю за збором та проходженням платежів за житлово-комунальні послуги та для забезпечення можливості оплати за надані комунальні послуги без простоювання у чергах.

Міська автоматизована система прийому та обліку комунальних платежів ґрунтується на наступних принципах:

- єдина міська база платників комунальних послуг;
- кожній квартирі присвоєно універсальний платіжний код, що відповідає її адресі;
- інформація про нарахування та заборгованість по кожному платнику щомісяця, на 1 число поточного місяця, надається банківським установам для Пунктів Прийому Платежів (ППП);
- платнику в Пункті Прийому Платежів надається інформація про нарахування, пільги, субсидії, заборгованість та іншу інформацію, усно, а також у вигляді друкованої Довідки;
- платник, за своїм бажанням, може здійснити оплату за комунальні

послуги повністю або частково за будь-який період;

– щомісячно ГЕРЦ формує та виготовляє Рахунок-повідомлення за адресою кожного абонента.

Адресна доставка Рахунків – повідомлень виконується через абонентські служби житлових комунальних підприємств.

Автоматизована система передбачає прийом комунальних платежів населення без використання розрахункових книжок. Касир пункту прийому платежів за адресою абонента, за особовим рахунком або за банківською платіжною картою НСМЕП знаходить у системі інформацію про борги та нарахування та здійснює відповідну операцію з прийому платежу. Клієнту на руки видається квитанція про оплату встановленого зразка, затвердженого Національним банком України.

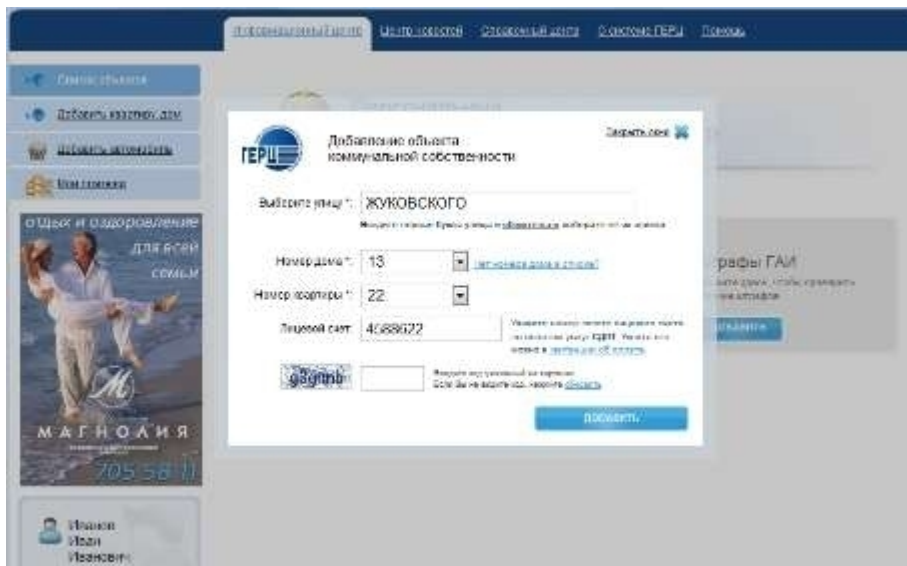


Рисунок 2.4 – Інтерфейс програми ГЕРЦ «Додавання об'єкта комунальної власності»

2.5 Комплексна автоматизована система «ЕЛЛІС-ЖКГ»

ЕЛЛІС-ЖКГ – сучасний програмний продукт, створений на основі системи управління базами даних MS SQL Server і призначений для автоматизації розрахунку квартплати, комунальних та інших платежів [7].

Основні відмінні риси ЕЛЛІС-ЖКГ:

- клієнт-серверна архітектура інформаційної мережі;
- доступ до бази даних у режимі реального часу необмеженої кількості користувачів;
- можливість обслуговування необмеженої кількості особових рахунків з веденням щодо кожного з них розгорнутої історії нарахувань та платежів за весь період роботи;
- електронний обмін інформацією центрального сервера з керуючою компанією, паспортним столом, службою соціального захисту населення, банками, відділеннями зв'язку, постачальниками послуг та іншими організаціями та підприємствами;
- можливість віддаленого адміністрування роботою програмно-апаратного комплексу;
- виконання нарахувань за алгоритмами будь-якого ступеня складності завдяки гнучкому налаштуванню розрахункової частини програми (вбудована мова програмування);
- програмний захист бази даних, детальне розмежування прав доступу користувачів до компонентів програми та бази даних, збереження відомостей про користувачів та про всі зміни, вчинені ними;
- коректний супровід необмеженої кількості загальних, колективних та індивідуальних приладів обліку (лічильників);
- використання технології штрихового кодування документів;
- можливість розширення системи за рахунок підключення до неї нових програмних модулів.

2.6 Система ISpro

Підсистема обліку комунальних послуг (ЖЕК/ОСББ) системи ISpro дозволяє здійснювати завдання обліку та управління даними про нарахування квартирної плати, комунальних послуг у житлово-експлуатаційних конторах (ЖЕК) та об'єднання співвласників багатоквартирних будинків (ОСББ). За

допомогою підсистеми проводиться [8]:

- ведення реєстру будинків та квартир;
- облік мешканців;
- Облік особових рахунків квартир (договорів);
- щомісячне виставлення квитанцій на оплату комунальних послуг;
- облік показників лічильників;
- облік оплати послуг;
- формування необхідної звітності.

Підсистема обліку комунальних послуг включає модулі:

- налаштування;
- особові рахунки ЖЕК (договору);
- будинки;
- квартири;
- реєстрація населення;
- рахунки за комунальні послуги;
- оплата комунальних послуг;
- номенклатура;
- прайс-листи;
- показання лічильників;
- звіти;
- довідник ВТН;
- журнал обліку господарських операцій;
- відомості списання;
- акти-претензії;
- імпорт показників лічильників;
- імпорт субсидій;
- імпорт пільговиків;
- закриття періодів.

Головне меню підсистеми наведено на рисунку 2.5.

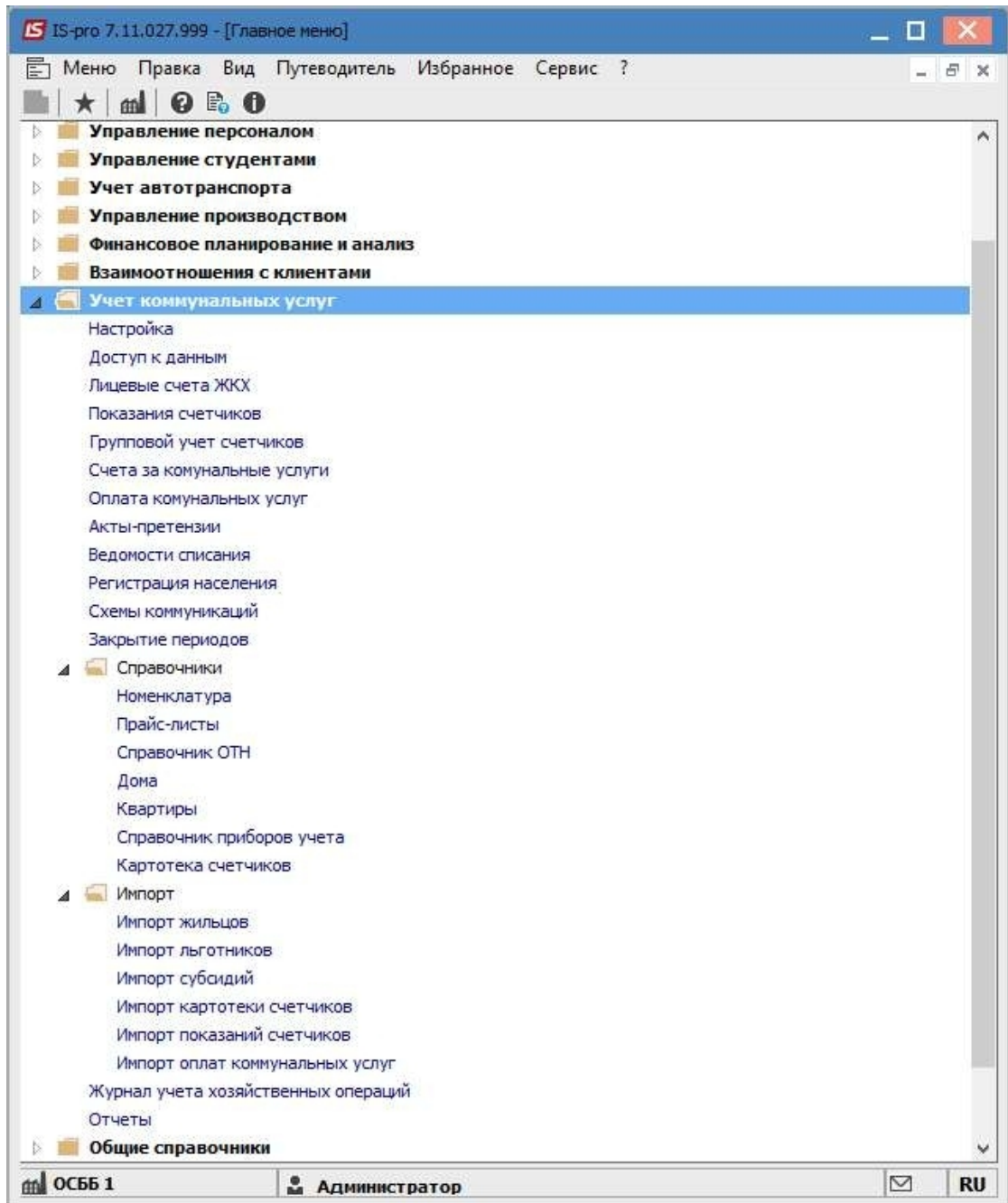


Рисунок 2.5 – Головне меню підсистеми обліку комунальних послуг

Для роботи у підсистемі необхідно внести таку інформацію:

- заповнити у Довіднику адрес (підсистема Загальні довідники) дані по своєму місту, вулиці;
- заповнити довідник Картка підприємства;
- внести до довідника Контрагентів дані щодо мешканців. Якщо необхідно формувати якісь списки по мешканцям, то вносяться всі, хто живе в будинку (не тільки платники послуг). Якщо такої потреби немає, то вноситься

лише власники особових рахунків та пільговики. Рекомендується створити групу контрагентів Мешканці. Для зручності роботи можна створити групи для кожного під'їзду або згрупувати за якоюсь іншою ознакою. Інформація, розміщена в цьому довіднику, використовується і при рознесенні оплати та в інших випадках (наприклад, формування списків мешканців за якимись ознаками тощо);

- внести дані по дому (модуль Будинку);
- внести дані про квартиру (модуль Квартири);
- сформувати реєстр особових рахунків (модуль Особові рахунки).

Технологія заповнення має на увазі заповнення даних у порядку їх перерахування.

Дані щодо пільг при оплаті послуг коригуються лише за зміни законодавства (модуль Налаштування) або при додаванні нової послуги до довідника номенклатур, за якою надаються пільги.

2.7 Бездротова мережева технологія LoRa

З активним використанням Інтернету речей з'являється все більша потреба в енергоефективних мережах. Багато програм Інтернету речей вимагають, щоб дані збиралися і відправлялися на великі відстані, причому датчики іноді знаходяться на відстані кількох кілометрів один від одного. Крім того, для розгортання Інтернету речей, можуть знадобитися пристрої для періодичної передачі невеликих пакетів інформації протягом багатьох років або навіть десятиліть.

Для забезпечення таких мереж з'явилася технологія LPWAN – енергоефективна мережа далекого радіусу дії. Передбачається, що глобальні мережі з низьким енергоспоживанням (LPWAN) підтримуватимуть більшу частину мільярдів пристроїв, які прогнозуються для Інтернету речей (IoT) . LPWAN мережа може базуватися на одній із технологій LoRaWAN, SIGFOX, NB-IoT, Weightless P тощо. Термін LoRaWAN означає протокол зв'язку та системну

архітектуру мережі, а фізичний рівень LoRa - модуляція, що забезпечує зв'язок на велику відстань. Саме LoRaWAN найбільше впливає на термін служби батареї вузла, ємність мережі, якість обслуговування, безпеку та різноманітність додатків, які обслуговує мережу[9].

Архітектура мережі LoRaWAN наведена на рисунку 2.6.

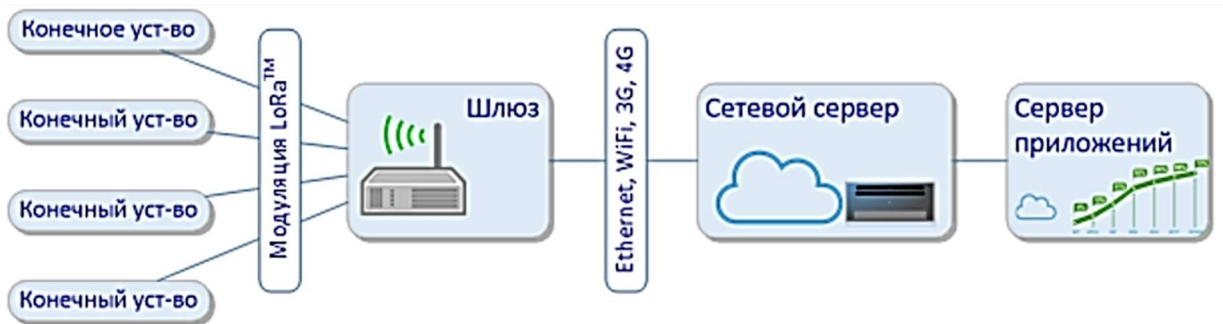


Рисунок 2.6 – Архітектура мережі LoRaWAN

Мережа LoRaWAN складається з наступних елементів: кінцевий пристрій, шлюзи, мережевий сервер та сервер додатків.

Кінцевий пристрій призначений для здійснення керуючих або вимірних функцій. Містить набір необхідних датчиків та керуючих елементів.

Шлюз – пристрій, що приймає дані від кінцевих пристроїв за допомогою радіоканалу та передає їх у транзитну мережу. Як транзитна мережа можуть виступати мережа Ethernet, WiFi або мережі рухомого радіотелефонного зв'язку. Шлюз та кінцеві пристрої утворюють мережеву топологію типу зірка. Зазвичай даний пристрій містить багатоканальні приймачі для обробки сигналів в декількох каналах одночасно або навіть декількох сигналів в одному каналі. Відповідно, декілька таких пристроїв забезпечує зону радіопокриття мережі та прозору двонаправлену передачу даних між кінцевими пристроями та сервером.

Мережевий сервер - призначений для управління мережею: завданням розкладу, адаптацією швидкості, зберіганням та обробкою даних, що приймаються.

Сервер додатків може віддалено контролювати роботу кінцевих пристроїв і збирати необхідні дані з них.

Зрештою, LoRaWAN мережа має топологію зірка зі зірок, має кінцеві пристрої, які через шлюзи, що утворюють прозорі мости, спілкуються із центральним сервером мережі. За такого підходу зазвичай передбачається, що шлюзами і центральним сервером володіє оператор мережі, а кінцевими пристроями – абоненти. Абоненти мають можливість прозорої двонаправленої та захищеної передачі даних до кінцевих пристроїв.

Переваги LoRaWAN:

- велика дальність передачі радіосигналу в порівнянні з іншими бездротовими технологіями, що використовуються для телеметрії, сягає 10...15 км;
- низьке енергоспоживання кінцевих пристроїв, завдяки мінімальним витратам енергії на передачу невеликого пакету даних;
- висока проникаюча здатність радіосигналу в міській забудові під час використання частот субгігагерцового діапазону;
- висока масштабованість мережі на великих територіях;
- відсутність необхідності отримання частотного дозволу та плати за радіочастотний спектр, внаслідок використання неліцензованих частот (ISM band).

Недоліки LoRaWAN:

- відносно низька пропускна здатність, що варіюється в залежності від використовуваної технології передачі даних на фізичному рівні, становить від декількох сотень біт/с до декількох десятків кбіт/с;
- затримка передачі даних від датчика до кінцевого додатка, пов'язана з часом передачі радіосигналу, може досягати від кількох секунд до кількох десятків секунд;
- відсутність єдиного стандарту, який визначає фізичний шар та керування доступом до середовища для бездротових LPWAN-мереж;
- ризики шуму спектру неліцензованого діапазону частот;
- пропрієтарна технологія модуляції LoRa, "закрита" патентом Semtech;
- обмеження потужності сигналу.

2.8 Значення сучасних IoT в сьогодні

Концептуально хмара є модель інтегрованого та спільного використання технологічних ресурсів, запропоновану постачальниками на ринку.

На практиці IT-інфраструктура, так і програмне забезпечення пропонується на віртуальних платформах третіми сторонами, а компанії-клієнти укладають контракти на послуги, такі як зберігання або обробка даних. Іншою можливістю є використання онлайн-систем, що замінюють необхідність внутрішньої розробки чи придбання рішень.

Тобто, “хмари” це ресурс, за допомогою якого можливо розгортати великі інформаційні системи використовуючи хмарні сховища, для зручності та швидкості оброблення інформації. Нижче, на рисунку 2.7, зображено, що може обчислювати за допомогою хмарних обчислень, розміщувати дані з ПК, персональних телефонів, великі бази даних, сервери, та обладнання управління “розумного дому”.

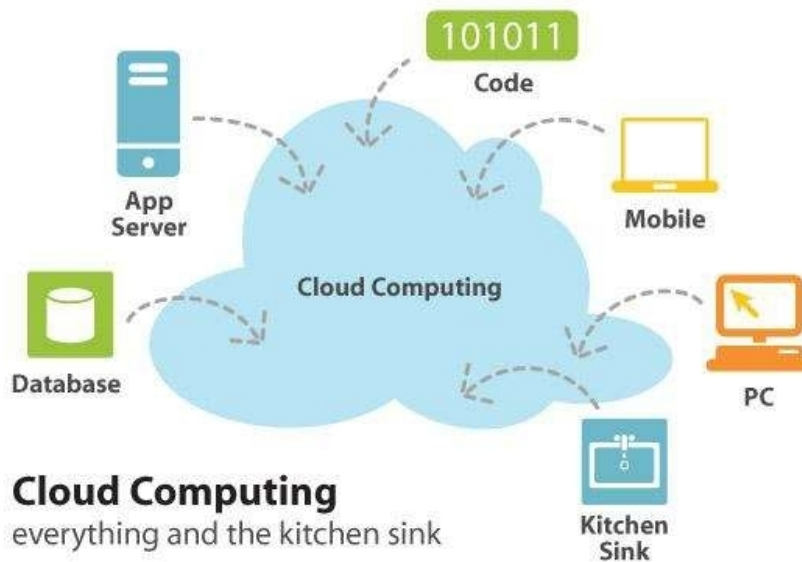


Рисунок 2.7 – Хмарні обчислення

Хмарні обчислення та використання хмар значно спрощує існування та налагодження інформаційних систем. У цієї технології є ряд переваг, чим не можуть похизуватися навіть найпотужніші комп’ютери.

Переваги, які отримує компанія що, вирішує використовувати хмарні обчислення:

- спільна робота: після інтеграції систем та доступу до онлайн-платформ спільна робота стає простішою;
- інформація доступна в режимі реального часу: оновлення даних відбувається негайно на серверах провайдера, що гарантує надійність інформації, що сприятиме ухваленню стратегічних рішень;
- інформаційна безпека. Аутсорсинг ІТ-послуг забезпечує безпеку бізнес-даних. Провайдери постійно оновлюють корпоративні антивіруси, брандмауери та механізми шифрування, щоб забезпечити цілісність інформації, що проходить через їхні центри обробки даних;
- резервне копіювання: турбота про копіювання файлів та управління процедурами резервного копіювання стає непотрібною, оскільки постачальник бере на себе цю відповідальність та гарантує швидке відновлення даних у разі інциденту;
- управління знаннями: поширення інформації та досвіду отримує стимул, оскільки дані та документи централізовані в середовищі, доступному для співробітників компанії.

Тож, можу зробити висновок, що хмарні обчислення значно прискорюють та дають змогу налагодити процеси, щоб інформаційна система, яка описується в даній кваліфікаційній роботі працювала правильно, без помилок та обробляла інформацію без затримок.

3 СХЕМОТЕХНІЧНА РОЗРОБКА

3.1 Розробка структурної схеми пристрою

Структурна схема пристрою для отримання показників електроенергії та витрат води у користувача, та відправки отриманих даних до серверу для подальшої обробки показників комунальних послуг наведена на рисунку 3.1

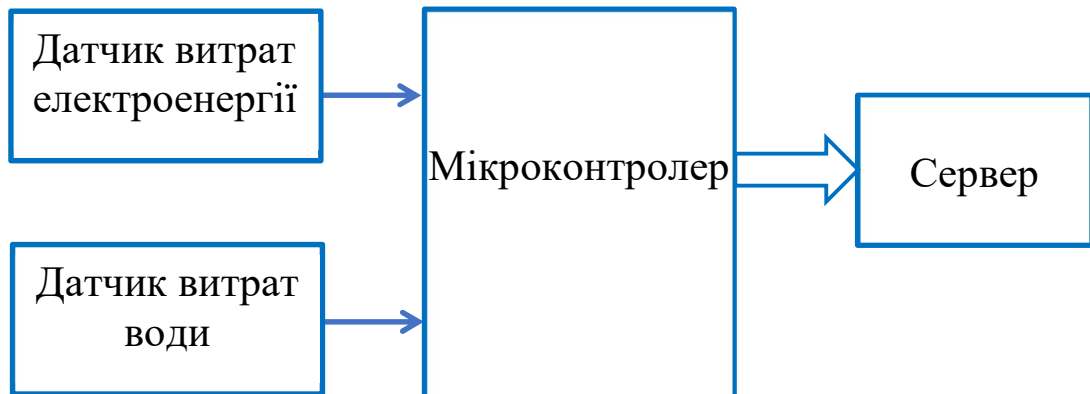


Рисунок 3.1 – Структурна схема пристрою

Відповідно до завдання, пристрій повинен забезпечити контроль витрат електроенергії та води. З цією метою в схемі використовуються відповідні датчики.

Інформація з датчиків поступає на мікроконтролер в якому проводиться первинна обробка та підготовка даних для передачі на сервер в якому зберігається програма їх обробки.

3.2 Вибір датчика витрати електроенергії

3.2.1 Багатофункціональний вимірювач електроенергії PZEM-004T

Цифровий багатофункціональний модуль призначений для моніторингу змінного струму, напруги, потужності та лічильник спожитої енергії. Базується на мікросхемі V9881D від китайської компанії SDIC Microelectronics. Окрім того, на платі є мікросхема EEPROM Atmel 24C02C на 256 байт з

мільйоном циклів запису і дві оптопари PC817, що забезпечують гальванічну розв'язку послідовного інтерфейсу UARTу, по якому передаються показники від модуля [10].

Зовнішній вигляд модуля наведено на рисунку 3.2.



Рисунок 3.2 – Зовнішній вигляд модуля PZEM-04

Основні характеристики PZEM модуля:

- діапазон вимірювання потужності споживання: 0 ~ 22 кВт;
- діапазон вимірювання напруги змінного струму: 80 ~ 260 В;
- діапазон вимірювання струму: 0 ~ 100А;
- робоча частота: 45-65 Гц;
- інтерфейс зв'язку: UART;
- напруга живлення модуля: 5 В;
- точність вимірювання: 1% похибки.

Серед основних можливостей та функцій даного модуля є:

- вимірювання електричних параметрів (напруга, струм, потужність, лічильник кВт/год);
- кнопка скидання показників;
- вимкнення живлення;

- робота з різними термінальними програмами, відображення даних на ПК (моніторинг напруги, струму, активної потужності);
- функція передачі даних з послідовним інтерфейсом.

Необхідно зауважити, що цей модуль призначений для використання в закритому корпусі, а не у відкритому вигляді, так як гальванічну розв'язку має лише інтерфейс UART, а решта елементів в тій чи іншій мірі пов'язані з мережею.

Приклад підключення до Wi-Fi модуля наведено на рисунку 3.3.

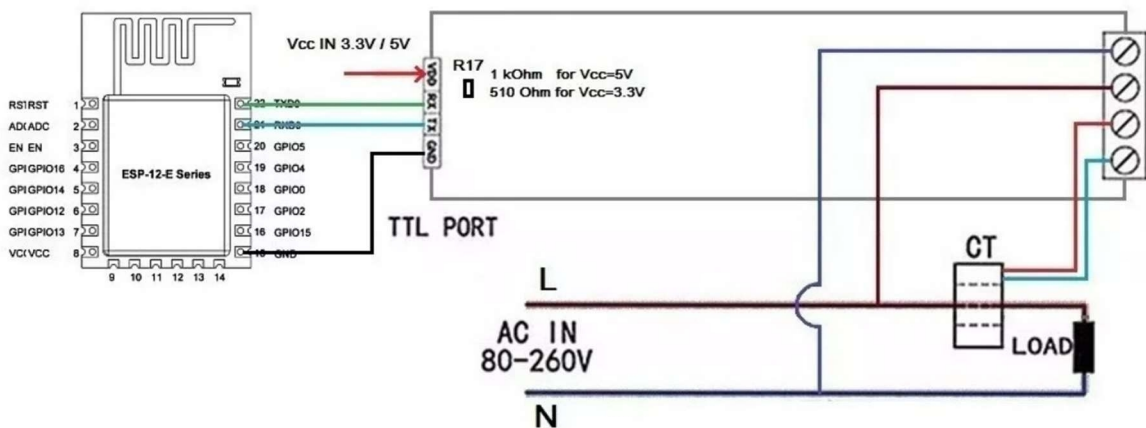


Рисунок 3.3 – Приклад підключення модуля до Wi-Fi

3.2.2 Ватметр PeaseFair PZEM-022

Загальний вигляд ватметра наведено на рисунку 3.4.



Рисунок 3.4 – Загальний вигляд ватметра

До особливостей даного ватметра можна віднести [11]:

- функція вимірювання електричних параметрів (напруження, ток, активна потужність, енергія, частота, коефіцієнт потужності);

- функція сигналізації перевантаження (обробка сигналізації при підвищенні порога, мигання підсвітки);
- регулювання порогу обробки сигналізації (можна встановити порог аварійного сигналу вручну);
- збір показників з допомогою клавіші;
- збереження даних при виключеному питанні;
- великий екран LCD;
- функція підсвітки;
- функція блокування.

Параметри ватметра:

- робоча напруга: 80...260 В змінного струму;
- номінальна потужність: 100А/22000 Вт
- робоча частота: 45...65 Гц
- точність виміру: 1,0 клас

Схема підключення ватметра наведена на рисунку 3.5.

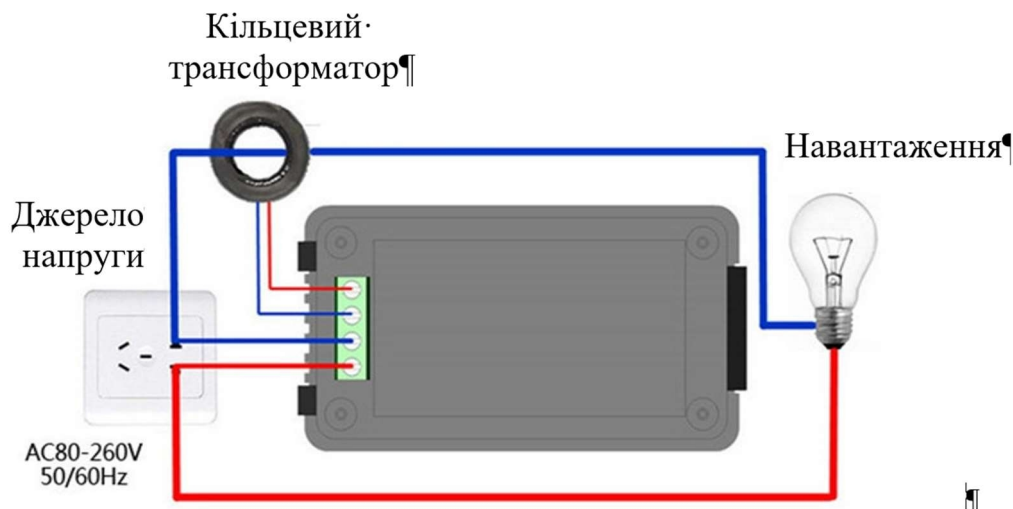


Рисунок 3.5 – Схема підключення ватметра

Аналіз розглянутих сенсорів вимірювання витрат електроенергії показав доцільність використання в проектованому пристрої багатофункціонального вимірювача електроенергії PZEM-004T приклад використання якого наведено на рисунку 3.6.

Принцип роботи датчика: при проходженні води через ротор починає обертатися. Швидкість обертання змінюється залежно від швидкості потоку води, і датчик Холла видає імпульсний сигнал відповідної частоти. Підключення до реєструючого пристрою здійснюється за допомогою трьох проводів: чорний – земля, червоний – живлення та жовтий – сигнальний [12].

Характеристики датчика:

- діаметр різьблення вхідного та вихідного штуцерів становить 1/2 дюйма;
- робоча напруга: 5...24 В;
- максимальний струм: 15 мА (5 В);
- діапазон витрати потоку: від 1 до 30 л/хв.;
- робочий тиск: до 1,75 МПа;
- температура рідини: до 120 °С;
- робоча температура: 0...80 °С;
- робоча вологість: 35...90 %.

3.3.2 Сенсор потоку YF-B5

Сенсор потоку YF-B5 це витратомір на основі датчика Холла. Потік води розкручує внутрішній ротор, що приводить у рух датчик Холла, який видає відповідний імпульсний сигнал [13].

Загальний вигляд сенсора YF-B5 наведено на рисунку 3.8.



Рисунок 3.8 – Загальний вигляд сенсора YF-B5

Характеристики сенсора:

- напруга живлення: 5...15 В;
- швидкість потоку: 1...30 л / хв;
- діаметр різьблення G3/4;
- частота імпульсів: F (Гц) = $6.6 * Q$ ($Q =$ л/хв);
- робоча температура: ≤ 80 °С;
- температура рідини: ≤ 120 °С;
- робоча вологість: 35...90 % RH;
- тиск води: ≤ 1.75 МПа;
- вихідний сигнал: імпульсний.

3.3.3 Ультразвуковий датчик витрати води Belimo FM0100F-SZ

Датчик призначається вимірювання витрати рідких середовищ. Ідеально підходить для систем кондиціонування, опалення, вентиляції, а також для застосування у промислових системах HVAC. Може використовуватися як у технологічних процесах, так і в обліково-розрахункових операціях житлово-комунального господарства. Ультразвуковий витратомір оснащений витратною трубкою, 4-ма датчиками витрати і блоком електроніки [14].

Загальний вигляд датчика наведено на рисунку 3.9.



Рисунок 3.9 – Загальний вигляд датчика Belimo FM0100F-SZ

Технічні характеристики:

- підключення: кабель 1 м, 3 x 0.75 мм²;
- допустимий тиск, ps: 1600 kPa;
- трубне з'єднання: фланець PN 16 згідно з EN 1092-2;
- клас захисту IEC/EN: III клас;
- позиція установки: пряма до горизонталі;
- точність вимірювання: ±2 %;
- температура навколишнього середовища: -30...50 °C;
- температура рідини: -20...120 °C.

3.3.4 Датчик витрат води DN15

Загальний вигляд датчика наведено на рисунку 3.10 [15].



Рисунок 3.10 – Загальний вигляд датчика DN15

Характеристики датчика:

- діапазон вимірювання 0,2...3,0 м³/год;
- вихідний сигнал 4...20 mA;
- тиск середовища 12/6 бар (при 40°C/100°C);

- діаметр різьблення G3/4.

Проведений огляд та аналіз датчиків витрат води показав доцільність використання в проектуваному пристрої датчика G 1/2.

3.4 Вибір мікроконтролера

Плата з мікроконтролером ESP32 використовується для отримання витрат споживаної електроенергії за допомогою модулю PZEM та отримання показників витрат води за допомогою датчику витрат води, після чого отримані дані надсилаються до серверу для подальшої обробки та зберігання показників комунальних послуг користувача. На рисунку 3.11 приведено зовнішній вигляд плати ESP32.

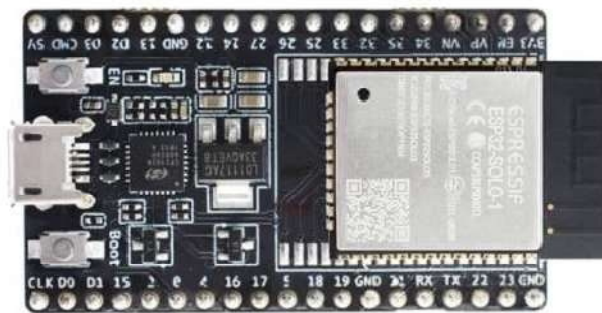


Рисунок 3.11 – Зовнішній вигляд плати з мікроконтролером ESP32

Основні характеристики плати ESP32:

- двох-ядерний 32-бітний процесор;
- об'єм SRAM пам'яті 520 Кб;
- максимальний струм споживання плати 260 мА, мінімальний струм споживання плати – 10 мА;
- стандарти підтримки безпроводного зв'язку: Wi-Fi: 802.11 b/g/N, Bluetooth: v4.2 BR/EDR and BLE;
- потри з підтримкою ШІМ;
- потри з підтримкою I2C інтерфейсу;
- потри з підтримкою UART інтерфейсу;

- потри з підтримкою SPI інтерфейсу;
- потри з підтримкою АЦП;
- потри з підтримкою ЦАП;

Розташування виводів залежить від виробників, плата DevKit з мікроконтролером ESP32 має 36 контактів. Функціональність портів вводу/виводу плата з мікроконтролером ESP32 приведено на рисунку 3.12.

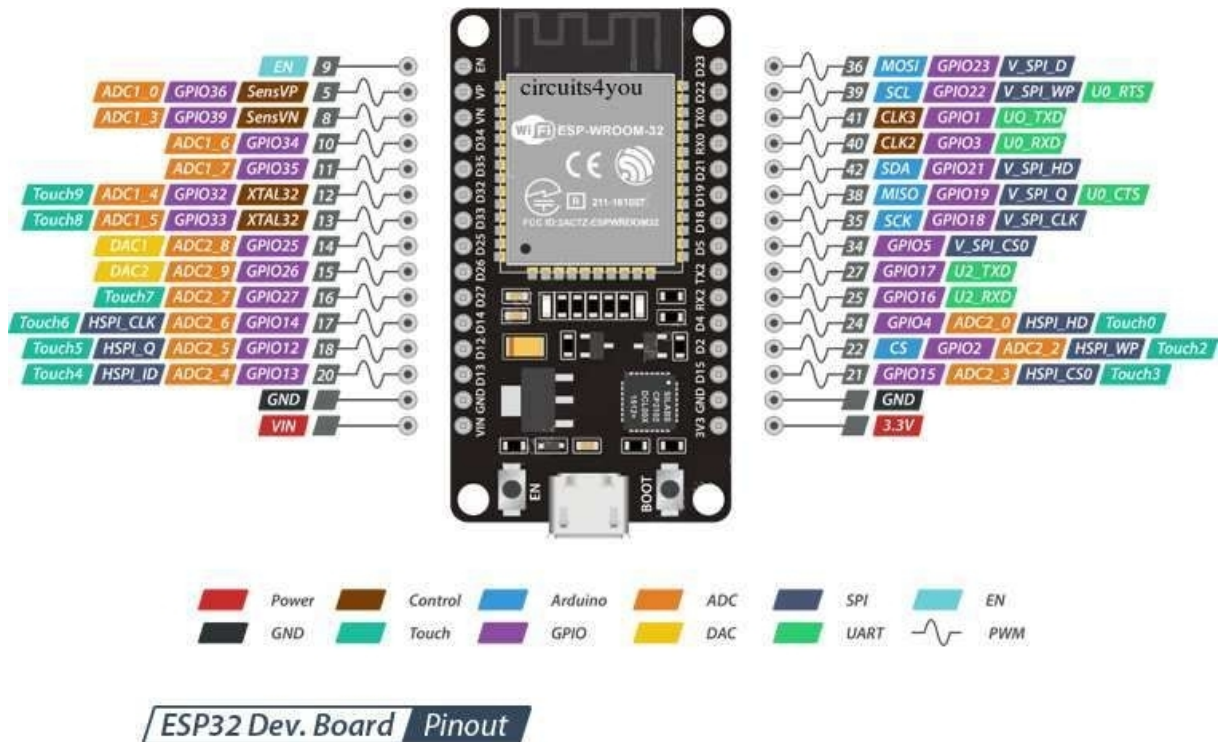


Рисунок 3.12 – Функціональність портів плати з мікроконтролером ESP32

4 ПРОГРАМНА РОЗРОБКА

4.1 Розробка програми роботи мікроконтролера

Розробка коду проводилась в середовищі Arduino IDE версії 2.2.1. Середовище розробки було налаштовано для роботи з платою з мікроконтролером ESP32 та додатково встановлені бібліотеки для роботи з модулем PZEM, повний код програми наведено в додатку А.

Програма складається з основного файлу «system» та додаткових допоміжних файлів в яких описані функції для роботи з бездротовою мережею, отримання даних показників електроенергії використаної користувачем та води, перелік файлів приведено на рисунку 4.1. Опис функцій в окремих файлах виконано зменшення коду в основній частині програми та для зручності читабельності коду.

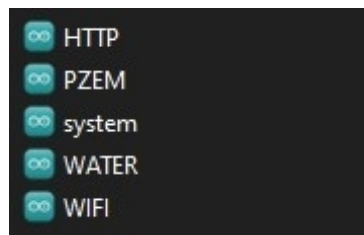


Рисунок 4.1 – Перелік файлів програмної частини

В файлі «system» знаходиться код основної програми в якому підключаємо всі необхідні бібліотеки, виконуємо налаштування портів вводу/виводу та налаштування інтерфейсів зв'язку як приклад в нашому випадку UART інтерфейс для зв'язку з модулем PZEM.

Перелік бібліотек що використовуємо:

- `#include <WiFi.h>` – дану бібліотеку використовуємо для роботи з бездротовою мережею Wi-Fi, що забезпечує нам канал передачі даних до серверу;
- `#include <ArduinoJson.h>` – використовуємо для роботи з JSON форматом даних що отримуємо від серверу для встановлення початкових показників електроенергії та води, також використовуємо даний формат для передачі

даних до серверу;

- `#include <HTTPClient.h>` – використовуємо для використання HTTP протоколу для відправки та отримання даних до серверу;

- `#include <PZEM004Tv30.h>` – використовуємо для роботи з модулем PZEM для отримання даних показників використаної електроенергії споживачем.

В основному файлі викликаємо наступні функції: `checkConectedToWifi`, `getDataPZEM`, `getWatermeter`, `HttpUpdate`.

Функція `checkConectedToWifi()` – дана функція описана в файлі `WiFi.ino`, дана функція виконує перевірку підключення до точки доступу у випадку відсутності зв'язку виконується повторне підключення до неї.

```
void checkConectedToWifi() {
  if ((WiFi.status() != WL_CONNECTED)) {
    Serial.println("Reconnecting to WiFi");
    WiFi.disconnect();
    WiFi.reconnect();
  } else {
    Serial.println("WiFi Connected");
  }
}
```

Функція `getDataPZEM()` – описана в файлі `PZEM.ino`, використовується для отримання даних з модуля PZEM. Також в даній функції виконується перевірка на коректність зчитування даних з модуля у випадку не коректного отримання даних виконується повторне зчитування даних з модуля.

Функція `getWatermeter()` – описана в файлі `Water.ino`, використовується для отримання даних про витрати води.

Функція `HttpUpdate()` – описана в файлі `HTTP.ino`, використовується для підготовки до відправлення даних про показників електроенергії та води в JSON форматі після чого відбувається відправка даних до серверу.

Сформований JSON має наступний вигляд:

```

{
  "state": {
    "reported": {
      "wifiRSSI": "-52",
      "sensors": {
        "energy": "2001",
        "water": "22",
      }
    }
  }
}

```

Відправка даних відбувається автоматично кожні 5 хвилин, при необхідності інтервал між відправками даних можна змінити в залежності від вимог замовника.

4.2 Розробка Web додатку

Розроблений пристрій для отримання показників комунальних послуг надсилає до розробленого нами веб додатку дані після чого вони зберігаються в базі даних, а через веб браузер є можливість візуально побачити ці дані. На рисунку 4.2 приведена структурна схема взаємодії Web Application з пристроєм для отримання даних комунальних послуг та браузером.

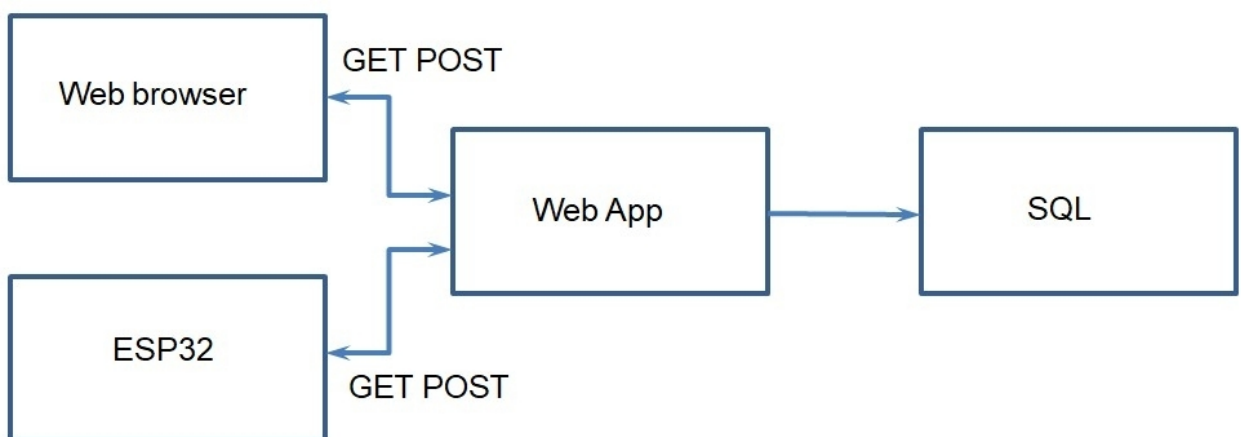


Рисунок 4.2 – Структурна схема взаємодії Web Application

Web Application був розроблений з використанням наступних фреймворків та інструментів: : Spring Boot, Spring Boot MVC, Liquibase, MapStruct, lombok, Bootstrap, Spring Data JPA та використовували базу даних MySQL.

Spring Boot - це фреймворк для розробки Java-додатків, який базується на платформі Spring Framework. Він надає простий та ефективний спосіб створення самостійних (standalone), готових до використання додатків, використовуючи конвенції за замовчуванням та автоматизовану конфігурацію.

Основні особливості Spring Boot включають:

- самостійність (Standalone): Додатки, створені з використанням Spring Boot, можуть бути запущені як самостійні Java-програми, не вимагаючи складної конфігурації або установки серверів додатків;

- автоматична конфігурація: Spring Boot надає розумні значення за замовчуванням та автоматичну конфігурацію на основі визначених конвенцій. Це допомагає уникнути багато рутинної роботи з налаштуванням;

- вбудований контейнер для веб-додатків: Spring Boot містить вбудовані контейнери для обробки HTTP-запитів, що робить простіше створення веб-додатків без додаткових конфігурацій;

- управління залежностями: Spring Boot використовує концепцію "starter" для автоматичного управління залежностями та надає зручний спосіб додавання необхідних бібліотек до проекту;

- спрощений доступ до баз даних, безпеки, легування та інше.

За допомогою Spring Boot нам як розробникам виникає можливість швидше створювати функціональні додатки, концентруючись на бізнес-логіці, а не на складних конфігураціях.

Spring Boot також включає в себе можливості для створення веб-додатків, та для цього використовується підтримка шаблону проекту Model-View-Controller (MVC).

Основні компоненти Spring Boot MVC включають в себе:

- контролери (Controllers): Контролери в Spring Boot MVC відповідають за обробку HTTP-запитів і визначають логіку, яка повинна бути виконана

при отриманні певного запиту. Контролер може повертати дані, які будуть використані для генерації відповіді;

- моделі (Models): Моделі представляють дані, які можуть бути використані у представленні (View). Це можуть бути об'єкти Java або інші типи даних;
- представлення (Views): Представлення визначають, як дані моделі повинні бути відображені. Вони можуть бути JSP (JavaServer Pages), Thymeleaf, FreeMarker, або іншими технологіями;
- конфігурація: Spring Boot автоматично конфігурує багато аспектів Spring MVC на основі конвенцій та значень за замовчуванням, але також надає можливість власної конфігурації через анотації або Java-код.

Основні переваги Spring Boot MVC:

- простота використання: Завдяки конвенціям та значенням за замовчуванням, Spring Boot MVC робить простіше розробку веб-додатків;
- швидкість розробки: Вбудовані засоби, такі як автоматична конфігурація та управління залежностями, дозволяють розробникам швидше створювати функціональні додатки;
- розширюваність: Легко розширювати функціональність за допомогою різноманітних розширень та додаткових бібліотек.

Загалом, Spring Boot MVC є потужним і зручним інструментом для розробки веб-додатків на основі Spring Framework.

Підчас розробки ми створили два контролери рисунок 4.3.

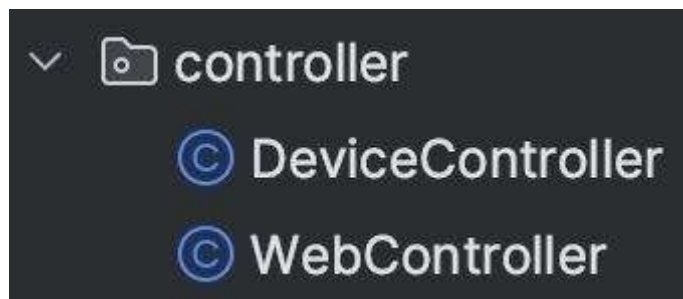


Рисунок 4.3 – Контролери Web Application

Зміст контролеру DeviceController приведено на рисунку 4.4.

```

@RequiredArgsConstructor
@RestController
@RequestMapping("/system")
public class DeviceController {
    private final DeviceService deviceService;

    @PostMapping(value = "/update/{id}")
    public DeviceDto updateDevice(@PathVariable Long id, @RequestBody DeviceRequestDto requestDto) {...}

    @GetMapping("/{id}")
    public DeviceDto getDeviceById(@PathVariable Long id) { return deviceService.getById(id); }
}

```

Рисунок 4.4 – Зміст контролеру DeviceController

Контролер DeviceController обробляє запити від ESP32, під час звертання POST запиту від ESP32 за адресою `system//update/{id}` додаток зберігає надіслані дані до бази даних. При звертанні GET запиту від ESP32 за адресою `/system/{id}` web додаток надсилає встановлені показники до ESP32, це необхідно для встановлення початкових показників від яких буде йти відлік.

Зміст контролеру WebController приведено на рисунку 4.5.

```

@GetMapping("/")
public String index() { return "index"; }

@GetMapping("/all-devices")
public String allDevices(Model model) {...}

@GetMapping("/device/add")
public String deviceAdd(Model model) { return "device-add"; }

@PostMapping("/device/add")
public String devicePostAdd(@RequestParam String address,
                             @RequestParam String flatNumber,
                             @RequestParam String energy,
                             @RequestParam String water,
                             Device device) {...}

@GetMapping("/device/edit")
public String deviceUpdate(Model model) {...}

```

Рисунок 4.5 – Зміст контролеру WebController

Контролер WebController оброблює запити з веб браузеру після чого в залежності від функції обробки ми відображаємо веб сторінку в браузері.

На рисунку 4.6 приведено відображення головної сторінки за запитом «/».

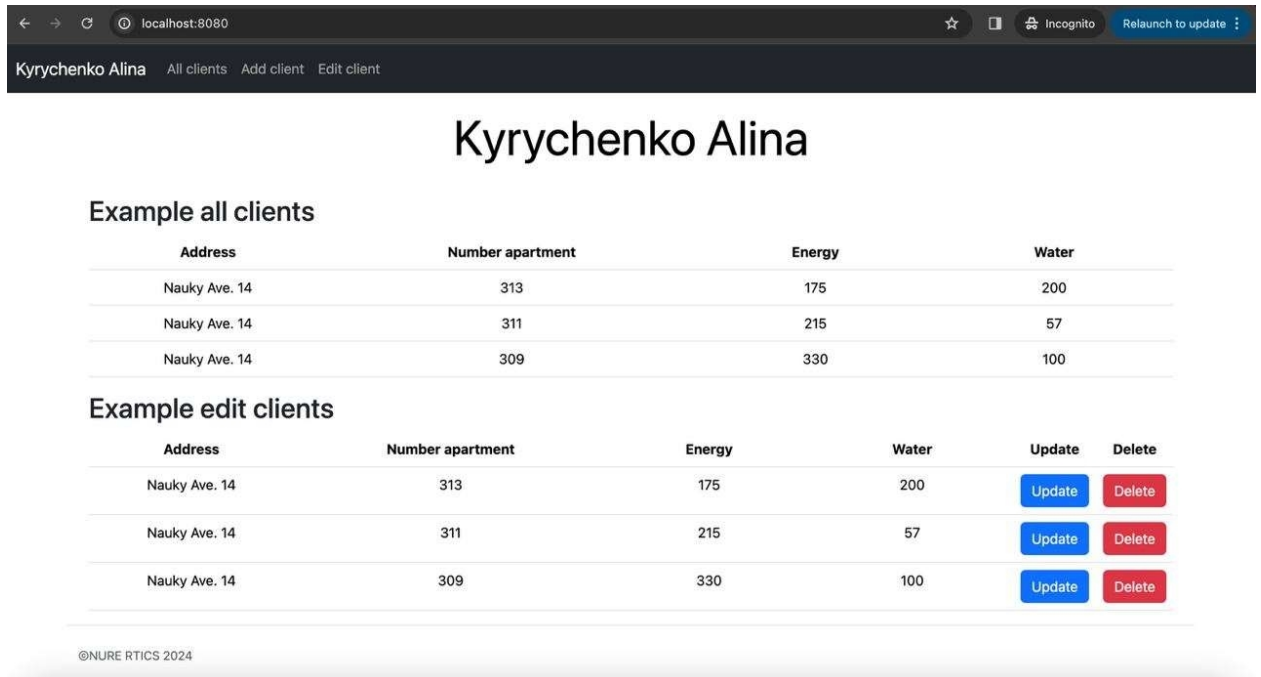


Рисунок 4.6 – Головна сторінка веб додатку

На головній сторінці приведено приклад відображення всіх клієнтів та сторінка редагування.

На рисунку 4.7 приведено сторінку відображення всіх користувачів з точними показниками комунальних послуг, адреса сторінки «/all-devices».

На рисунку 4.8 приведено відображення сторінки додати клієнта за посиланням /device/add, де вказуємо адресу користувача, номер квартири, поточні показники лічильника електроенергії, поточні показники лічильника води.

На рисунку 4.9 приведено відображення сторінки редагування запису клієнта, за посиланням /device/edit. Розроблено два режими редагування запису користувача це оновлення даних або видалення запису користувача.

У випадку кліку на кнопку «update» ми переходимо до сторінки оновлення записів користувача за посиланням /device/{id}/update, де ми можемо від коректувати всі поля користувача. Сторінка оновлення приведена на рисунку 4.10.

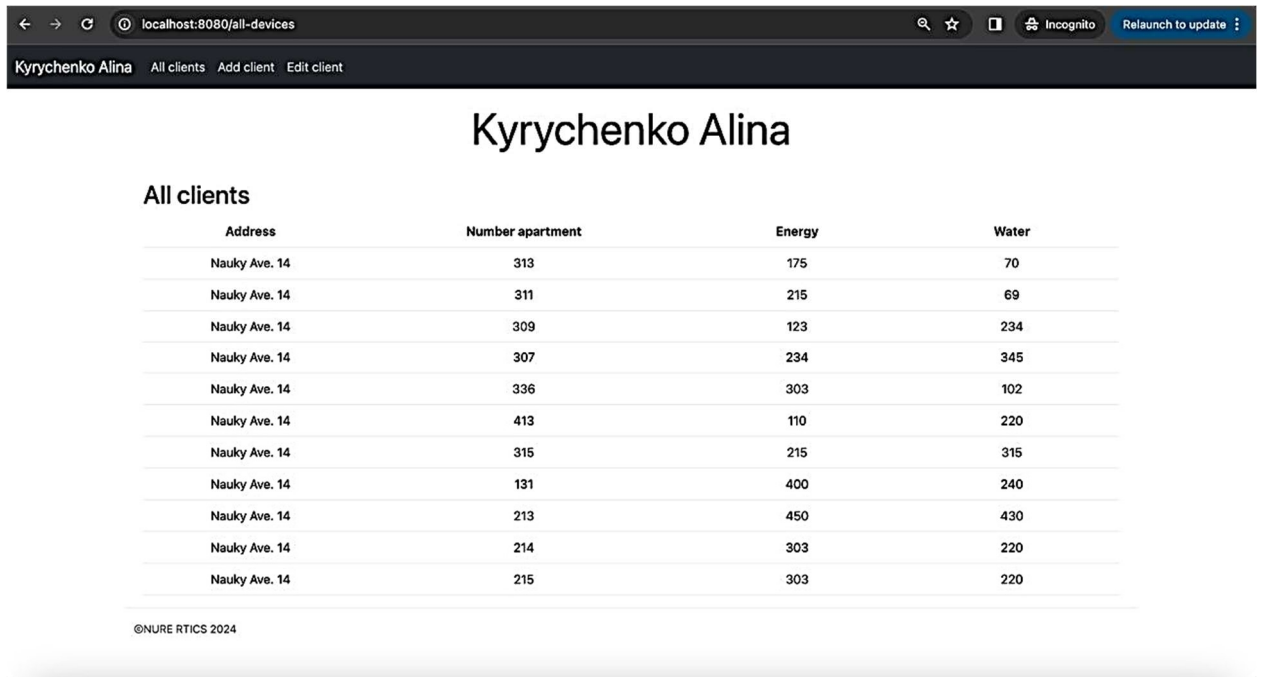


Рисунок 4.7 – Сторінка відображення всіх користувачів

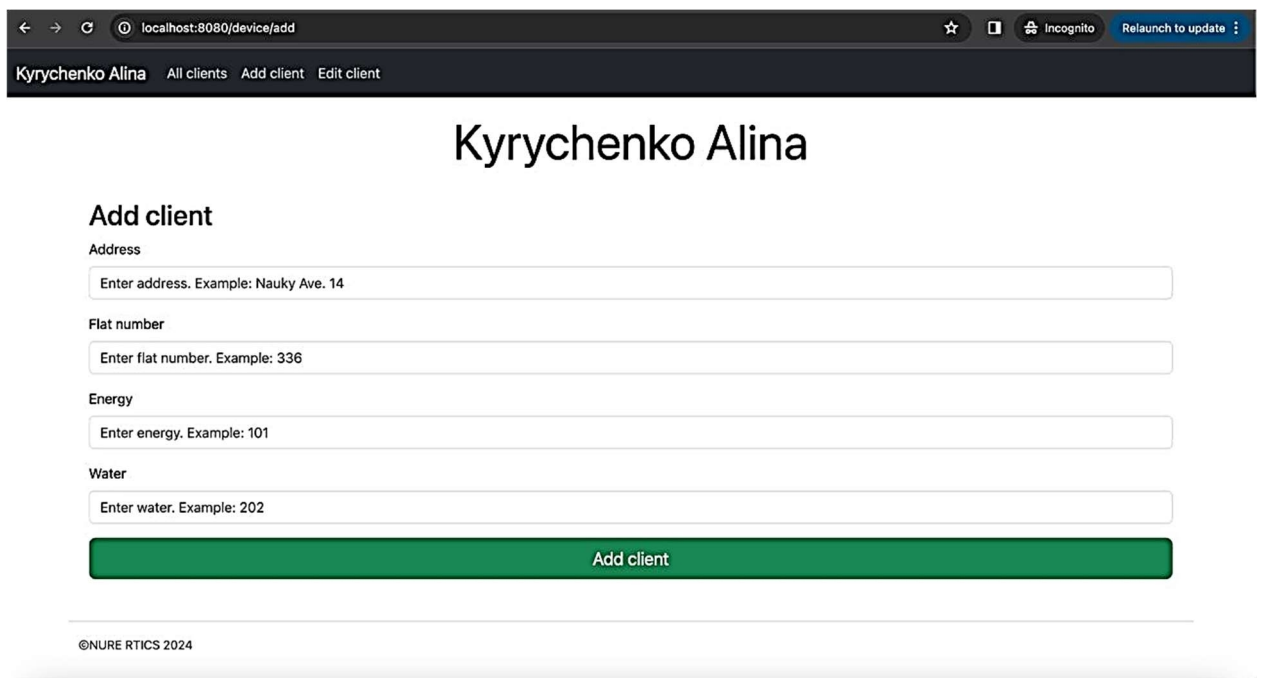


Рисунок 4.8 – Сторінка додавання нового користувача

У випадку кліку на кнопку «Delete» записи обраного користувача видаляються, після чого ми потравляємо на сторінку відображення всіх користувачів.

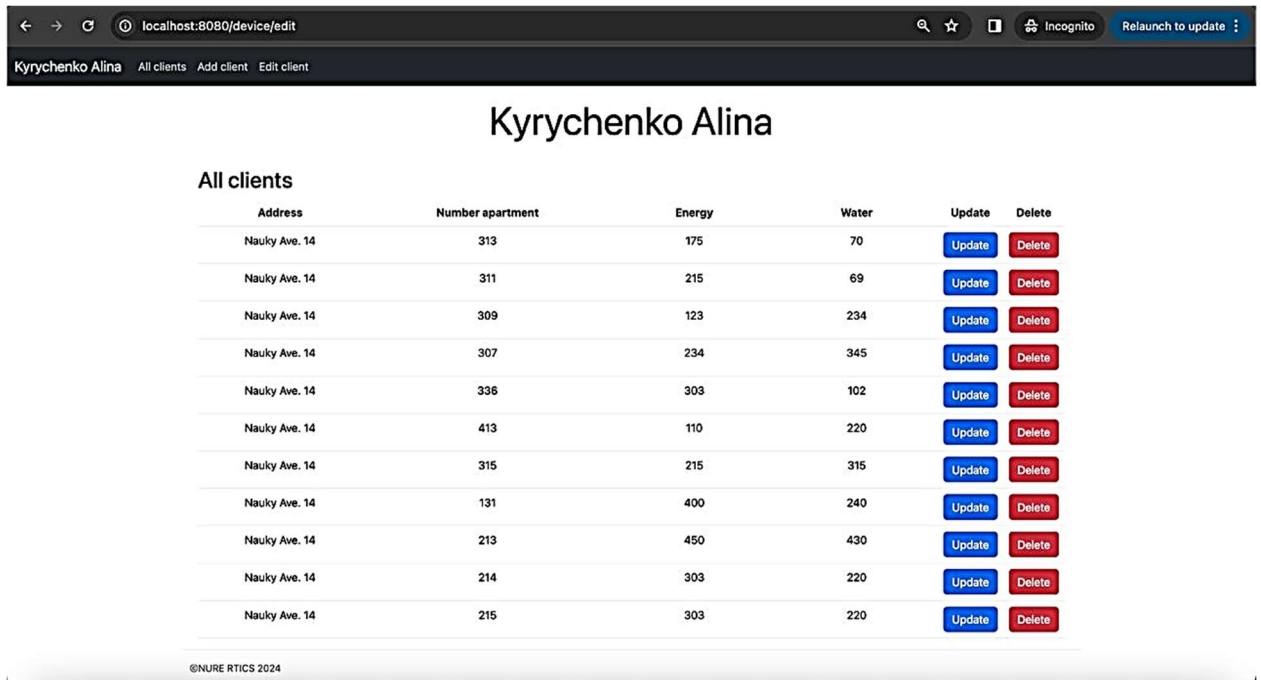


Рисунок 4.9 – Сторінка редагування користувача

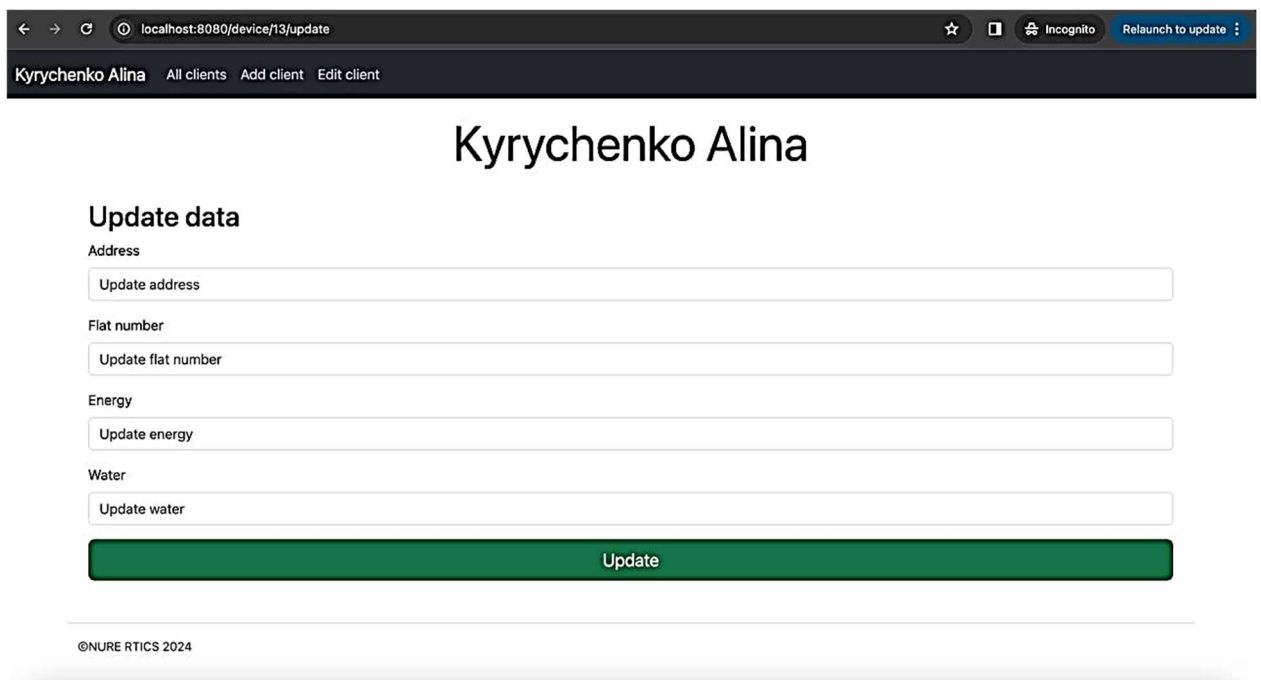


Рисунок 4.10 – Сторінка Оновлення даних користувача

Підчас розробки відображення сторінок використовували Bootstrap.

Bootstrap - це відкритий інструментарій для розробки веб-сайтів та веб-додатків. Він містить HTML, CSS і JavaScript-компоненти, які використовуються для створення сучасного та адаптивного інтерфейсу користувача.

Основні характеристики Bootstrap:

- готові компоненти: Bootstrap надає готові HTML- та CSS-компоненти, такі як кнопки, форми, таблиці, навігаційні панелі, модальні вікна та багато інших. Це дозволяє розробникам швидше створювати структуру та елементи веб-сайту;
- адаптивний дизайн: адаптивний дизайн, що дозволяє створювати веб-сайти, які гарно виглядають на різних пристроях та розмірах екранів;
- система сітки (Grid system): Bootstrap використовує 12-колонкову сітку, яка полегшує організацію та розміщення елементів на сторінці. Це дозволяє створювати гнучкі та ефективні макети;
- JavaScript-компоненти: Bootstrap містить JavaScript-компоненти, такі як висувні панелі, каруселі, модальні вікна, вкладки та інші, які додають динамічні функції до веб-сайтів.

На рисунку 4.11 приведено розташування та перелік всіх розроблених html веб сторінок.

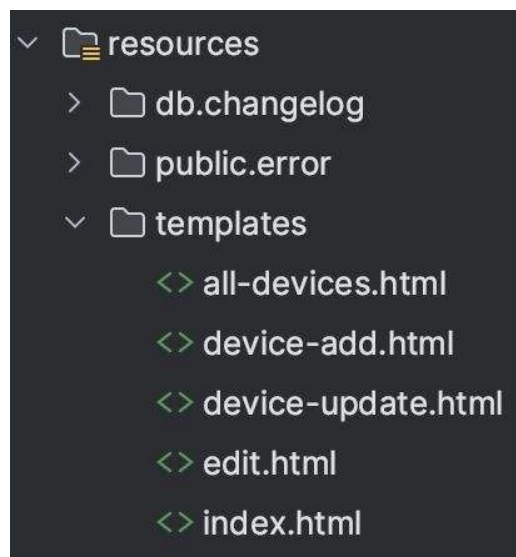


Рисунок 4.11 – Перелік html сторінок

При розробці веб сторінок також використовували Thymeleaf.

Thymeleaf - використовується для створення HTML, XML, JavaScript та інших текстових файлів. Він широко використовується в розробці веб-додатків на платформі Java, зокрема, в інтеграції з фреймворками веб-розробки, такими як Spring Framework.

Основні особливості Thymeleaf:

- синтаксис, легкий для сприйняття;
- інтегрований з HTML: Thymeleaf-теги можна вставляти в HTML-код, і вони будуть ігноровані браузером, якщо Thymeleaf не активований. Це робить шаблони Thymeleaf придатними для використання як звичайні HTML-файли;
- підтримка виразів: Thymeleaf використовує вирази, які дозволяють вбудовувати динамічні дані та логіку в шаблони. Вони виглядають подібно коду Java та можуть взаємодіяти з об'єктами, які передаються з сервера;
- підтримка форм та обробка подій: З Thymeleaf легко працювати з формами, обробляти події та валідувати дані;
- інтеграція з різними фреймворками: Thymeleaf часто використовується разом з різними фреймворками веб-розробки, такими як Spring Framework, для створення динамічних та привабливих веб-сторінок;
- підтримка шаблонів: Thymeleaf підтримує використання шаблонів, що дозволяє створювати заготовки та використовувати їх в різних частинах веб-сторінок.

При розробці Web Application також використовували Liquibase для створення таблиці та двох користувачів, рисунок 4.12.

Liquibase - це інструмент для керування версіями баз даних. Він дозволяє нам визначати та відстежувати зміни в структурі бази даних, а також автоматизує процес міграції бази даних між різними версіями програмного забезпечення.

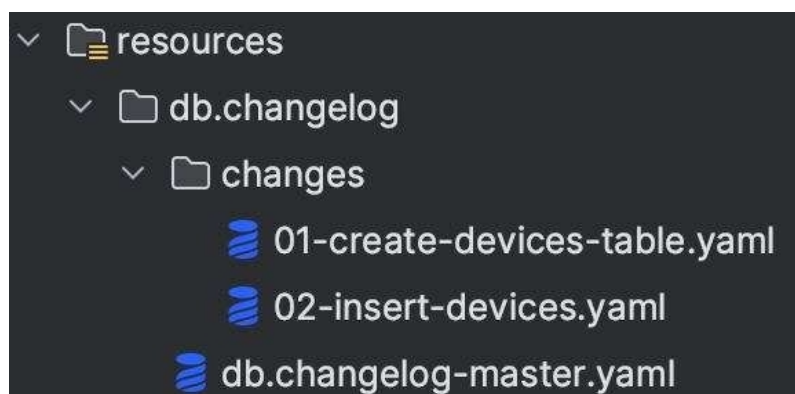


Рисунок 4.12 – Перелік html сторінок

Основні можливості Liquibase включають:

- декларативний підхід: Liquibase використовує декларативний підхід до визначення змін в базі даних. Розробники описують необхідні зміни за допомогою XML, YAML або SQL-файлів;
- крос-платформенність: Liquibase є крос-платформеним і може працювати з різними системами управління базами даних, такими як MySQL, PostgreSQL, Oracle, Microsoft SQL Server і багатьма іншими;
- контроль версій: Легко відстежувати та управляти версіями бази даних, зберігаючи історію змін та забезпечуючи можливість відкату до певної версії;
- автоматична міграція: Liquibase може автоматично виявляти зміни в базі даних і застосовувати їх при запуску додатка або сервісу;
- мульти-серверність: Здатність працювати в розподілених середовищах і керувати змінами в різних етапах розробки та випуску програмного забезпечення;
- робота з кодом та SQL: Ліквібейс підтримує зміни в базі даних, які можна визначити як кодом (на різних мовах програмування) чи SQL-скриптами;
- інтеграція з іншими інструментами: Liquibase легко інтегрується з різними інструментами для автоматизації процесу розробки та випуску.

Для зменшення кількості коду використовували бібліотеку Lombok.

Основні функції Lombok включають:

- анотації: Використання анотацій для додавання функціональності до коду без необхідності його явного написання. Наприклад, анотація `@Data` додає стандартні методи, такі як геттери, сеттери, `equals`, `hashCode` тощо;
- генерація методів: Lombok генерує код методів під час компіляції, замінюючи його на шляху до байт-коду. Це дозволяє уникнути дублювання коду та робить його більш компактним;
- спрощення конструкторів: Анотації, такі як `@AllArgsConstructor` або `@NoArgsConstructor`, автоматично генерують конструктори з усіма або

без аргументів відповідно;

- керування нульовими значеннями: Аннотація `@NonNull` дозволяє забезпечити, що вхідні параметри не можуть бути `null`;

- сумісність з іншими інструментами: Lombok легко інтегрується з іншими інструментами розробки, такими як редактори коду та інші фреймворки.

Для збереження отриманих даних клієнтів використовуємо базу даних MySQL. Оскільки вона є відкритою системою керування базами даних, яка використовує мову структурованих запитів SQL (Structured Query Language).

Основні характеристики MySQL:

- реляційна модель даних: MySQL використовує реляційну модель даних, що означає, що дані зберігаються у вигляді таблиць з рядками та стовпцями. Вона підтримує зв'язки між таблицями, індекси, транзакції та інші реляційні концепції;

- підтримка SQL: MySQL використовує мову запитів SQL для взаємодії з базою даних. SQL дозволяє виконувати операції вставки, оновлення, видалення та вибірки даних, а також визначати та змінювати структуру бази даних;

- підтримка транзакцій: MySQL підтримує транзакції, що дозволяє виконувати групу операцій як єдину атомарну одиницю. Це забезпечує консистентність бази даних у випадку помилок або відміни операцій;

- безпека та автентифікація: MySQL надає можливості для встановлення паролів, обмежень доступу та інших заходів безпеки для захисту даних в базі;

- широкі можливості індексації: Для оптимізації швидкості вибірки та пошуку даних, MySQL підтримує різні типи індексів;

- підтримка розподілених баз даних: MySQL Cluster надає можливість розподілити дані та навантаження на кілька серверів для високої доступності та масштабованості;

- спільнота та підтримка: MySQL є відкритим програмним забезпеченням, і його розвиток підтримується великою спільнотою розробників. Офіційний вендор також надає комерційну підтримку для корпоративних потреб.

Для відображення помилки 404 яка виникає у випадку звернення до неіснуючої сторінки ми розробили сторінку відображення помилки 404 сторінка не знайдена. Розміщення даної сторінки приведено на рисунку 4.13, а на рисунку 4.14 приведено зображення сторінки повідомлення про помилку сторінка не знайдена.

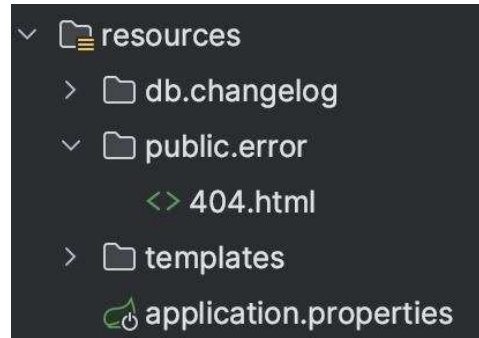


Рисунок 4.13 – Розміщення сторінки для відображення помилки 404

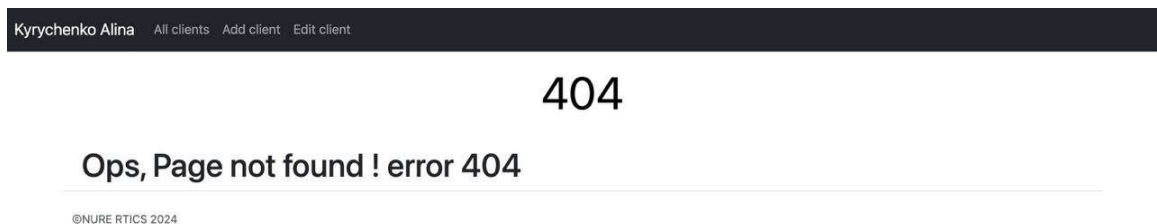


Рисунок 4.14 – Сторінка відображення помилки 404

Було розроблено пристрій для отримання показників комунальних послуг: використаної електроенергії та використаної води.

Початкові значення комунальних показників встановлюються через веб додаток, що спрощує встановлення пристрою для отримання даних показників. Пристрій для отримання показників розроблений на базі мікроконтролеру ESP32.

Для встановлення початкових параметрів на пристрої, ESP32 посилає запит до веб додатку за посиланням `/system/{id}`, після чого веб додаток у відповідь надсилає параметри в форматі JSON.

Пристрій після отримання показників використаної електроенергії та води надсилає до веб додатку за посиланням `/system /update/{id}` в форматі JSON дані.

ВИСНОВКИ

Метою магістерської кваліфікаційної роботи була розробка автоматичної системи збору даних комунальних послуг з розрахунком вартості за місяць.

В першому розділі роботи було розглянуто принципи побудови комерційного обліку та деякі аспекти впровадження автоматизованих систем.

В другому розділі роботи проведено огляд та аналіз аналогічних систем таких як система Tera Bills, автоматизована система управління підприємствами житлово-комунального господарства (АСУП ЖКГ) тощо.

При схемотехнічній розробці системи було проведено обґрунтування вибору мікроконтролера та обрано необхідні датчики.

В четвертому, програмному, розділі наведено програмні коди підключення датчиків до мікроконтролера та розроблено Web додаток.

Отже, всі поставлені задачі виконано в повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Закон України «Про комерційний облік теплової енергії та водопостачання» URL: <https://ukrvodokanal.in.ua/wp-content/uploads/2019/04/pro-komertsijnyj-oblik.pdf> (дата звернення 21.10.2023).
2. Закон України про житлово-комунальні послуги URL: <https://zakon.rada.gov.ua/laws/show/1875-15#Text> (дата звернення 21.10.2023).
3. Рыбцев І.В., Якубенко Я.В. Анализ автоматизированных систем учета оплаты коммунальных услуг URL: http://nvdu.snu.edu.ua/wp-content/uploads/2020/03/2012_8_24.pdf (дата звернення 25.10.2023).
4. Автоматизованая система “Комунальні платежі”. URL: <https://www.vas.com.ua/vas/opisan2.pdf>. (дата звернення 25.10.2023).
5. Tera Bills ver. 4.0 URL: https://filebox.ru/p/tera_bills/ (дата звернення 25.10.2023).
6. Автоматизация предприятий ЖКХ. Конфигурация "Водоканал, ЖЭК, ОСМД". Программа для бухгалтерского учета URL: <https://debet.com.ua/ru/avtomatizaciya-predpriyatij-zhkkh-konfiguraciya-vodokanal-zhek-osmd-programma-dlya-bukhgalterskogo> (дата звернення 25.10.2023).
7. Комплексная автоматизированная система «Эллис – ЖКХ URL:<http://erc-russia.ru> > site > page > url_label > ellis6 (дата звернення 25.10.2023).
8. Система ISpro. URL:<https://ispro.ua/uploads/usermanuals/November2020/mpNyYCbqbrG9EhetcHML.pdf>. (дата звернення 10.10.2023).
9. LoRaWAN. URL: <https://lanmarket.ua/ua/entsiklopediya/besprovodnye-tehnologii/lorawan.html>. (дата звернення 12.12.2023)
10. Монитор потребляемой электроэнергии 80-260V 100A PZEM-004T-100A V3.0. URL: <https://prom.ua/p1384126831-monitor-potrebljaemoj-elektroenergii.html> (дата звернення 12.12.2023)

11 Ватметр PZEM-22 URL: <https://radiostore.com.ua/ua/p975748738-vattmetr-pzem-schetchik.html>. (дата звернення 12.12.2023)

12 G 1/2 датчик расхода воды, счетчик, расходомер для Arduino. URL: https://evse.com.ua/g-1-2-datchik-rashoda-vody-schetchik-rashodomer-dlya-arduino?gclid=Cj0KCQiAhomtBhDgARIsABcaYynpaQDhI_uXKXhHTHZmH2HLuUBBR8mq-O2h9P-J81CfRwHASOACgwaAhInEALw_wcB (дата звернення 12.12.2023)

13 Сенсор потока YF-B5 3/4. URL: https://uamper.com/index.php?route=product/product&path=221&product_id=1436&gclid=Cj0KCQiAhomtBhDgARIsABcaYylpPQraITPQBpy2mn_VzldEbX9jMzIV8h3aFMGEOAqtTP0_v2fzKCIaAiGaEALw_wcB. (дата звернення 12.12.2023)

14 Ультразвуковой датчик расхода воды Belimo FM0100F-SZ | DN 100 | DC 0-10 В. URL: <https://profimann.com.ua/avtomatika-i-upravlenie/datchiki-rashoda-vody/ultrazvukovoy-datchik-rashoda-vody-belimo-fm0100f-sz-dn-100-dc-0-10-b/>(дата звернення 12.12.2023).

15 Датчики расхода воды DN15 с входным сигналом 4-20мА. URL: <https://enertec.ua/katalog-tovarov/datchiki-rashoda/datchiki-rashoda-vody-dn15-25-s-vhodnym-signalom-4-20ma1.html>. (дата звернення 12.12.2023)

Додаток А
(Рекомендований)

ПРОГРАМНІ КОДИ

Файл HttpGetData

```

void HttpGetData(){
    HTTPClient http;

    // Send request
    http.begin(GETDATA);
    http.GET();

    // Parse response
    DynamicJsonDocument doc(JSON_DOCUMENT_SIZE);
    deserializeJson(doc, http.getStream());

    // Read values
    setEnergy = doc["energy"].as<float>();
    setWater = doc["water"].as<float>();

    // Disconnect
    http.end();
}

void HttpUpdate() {
    DynamicJsonDocument stateDocument(JSON_DOCUMENT_SIZE);

    stateDocument[F("clientToken")] = THING_NAME;

    JsonObject state = stateDocument.createNestedObject(F("state"));
    JsonObject state_reported = state.createNestedObject(F("reported"));
    state_reported[F("wifiRSSI")] = WiFi.RSSI();
    JsonObject state_reported_sensors = state_reported.createNestedObject(F("sensors"));
    state_reported_sensors[F("energy")] = energyKW;
    state_reported_sensors[F("water")] = water;
    char buffer[SEND_MESSAGE_BUFFER_SIZE];
    serializeJson(stateDocument, buffer);
}

```

```
Serial.println("\n* SEND MSG TO SERVER ");

HTTPClient http;
http.begin(UPDATE);

// Specify content-type header
http.addHeader("Content-Type", "application/json");
http.POST(buffer);

//Print stateDocument in to Serial port
serializeJsonPretty(stateDocument, Serial);

//Read response
Serial.print(http.getString());

//Disconnect
http.end();

}
/*
{
  "state": {
    "reported": {
      "wifiRSSI": "-52",
      "sensors": {
        "energy": "2001",
        "water": "22",
      }
    }
  }
}
*/
```

Файл DataPZEM

```

void getDataPZEM(){
    float energy = pzem.energy();    // Energy    : kWh
    float voltage = pzem.voltage();   // Voltage   : V
    float current = pzem.current();   // Current   : A
    float power = pzem.power();       // Power     : W
    float frequency = pzem.frequency(); // Frequency : Hz
    float pf = pzem.pf();             // Power Factor

    if(!isnan(energy) || !isnan(voltage) ||
        !isnan(current) || !isnan(power) ||
        !isnan(frequency) || !isnan(pf)){

        Serial.println("Energy : " + String(energy) + " kWh");
        energyKW = energy + setEnergy;
        Serial.println("Voltage : " + String(voltage) + " V");
        Serial.println("Current : " + String(current) + " A");
        Serial.println("Power : " + String(power) + " W");
        Serial.println("Frequency : " + String(frequency) + " Hz");
        Serial.println("Power Factor : " + String(pf) + " PF");
        delay(2000);

        errorReadDataPZEM = false;
    }
    else {
        errorReadDataPZEM = true;
        Serial.println(" ERROR READ DATA FROM PZEM ");

        if(countErrorReadDataPZEM > 20) ESP.restart();
        else countErrorReadDataPZEM++;
    }
    delay(1500);
}

```

Файл Watermeter

```

void getWatermeter() {
currentMillis = millis();
if (currentMillis - previousMillis > interval) {

    pulse1Sec = pulseCount;
    pulseCount = 0;

    flowRate = ((1000.0 / (millis() - previousMillis)) * pulse1Sec) / calibrationFactor;
    previousMillis = millis();

    flowMilliLitres = (flowRate / 60) * 1000;

    totalMilliLitres += flowMilliLitres;

    Serial.print(int(flowRate)); // Print the integer part of the variable
    Serial.println("L/min");

    // Print the cumulative total of litres flowed since starting
    Serial.print(totalMilliLitres);
    Serial.print("mL / ");
    Serial.print(totalMilliLitres / 1000);
    Serial.println("L");

    waterL = (totalMilliLitres / 1000) + setWater;
}
}

```

Файл ConectedToWifi

```

void checkConectedToWifi() {
if ((WiFi.status() != WL_CONNECTED)) {
    Serial.println("Reconnecting to WiFi SSID: " + ssid);
    WiFi.disconnect();
}
}

```

```

    WiFi.reconnect();
  } else {
    Serial.println("WiFi Connected to: " + ssid);
  }
}

```

Файл system

```

#include <WiFi.h>

const char* ssid = "Kirichenko";
const char* password = "KirichenkoPassword";

#include <ArduinoJson.h>

#define JSON_DOCUMENT_SIZE 2048
#define SEND_MESSAGE_BUFFER_SIZE 2048
#define THING_NAME "hotel-number-1-room-number-1"

#include <HTTPClient.h>
#define GETDATA "http://system.info/get-data"
#define UPDATE "http://system.info/update"

#define PZEM_SERIAL Serial2
#include <PZEM004Tv30.h>
PZEM004Tv30 pzem(PZEM_SERIAL);
int countErrorReadDataPZEM = 0;

float energyKW = 0;
boolean errorReadDataPZEM = false;

#define WATERMETERPIN 2
float waterL = 0;
long currentMillis = 0;
long previousMillis = 0;
int interval = 1000;

```

```
boolean ledState = LOW;
float calibrationFactor = 4.5;
volatile byte pulseCount;
byte pulse1Sec = 0;
float flowRate;
unsigned int flowMilliLitres;
unsigned long totalMilliLitres;

void IRAM_ATTR pulseCounter()
{
  pulseCount++;
}

float setEnergy;
float setWater;

unsigned long lastTime = 0;
// Timer set to 5 minutes (300000)
unsigned long timerDelay = 300000;

void setup() {
  Serial.begin(9600);      // init Serial port
  WiFi.begin(ssid, password); //init WiFi

  // init pin for watermeter
  pulseCount = 0;
  flowRate = 0.0;
  flowMilliLitres = 0;
  totalMilliLitres = 0;
  previousMillis = 0;

  attachInterrupt(digitalPinToInterrupt(WATERMETERPIN), pulseCounter, FALLING);

  HttpGetData()           // get set param for energy and water
```

```
}  
  
void loop() {  
  checkConectedToWifi();      //check connection to WiFi  
  
  if((millis() - lastTime) > timerDelay){  
    getDataPEZEM();           //controle air contidions and ultraviolet lamp  
    getWatermeter();          //get parametrs for climat  
    HttpUpdate();             //send sensors data to server  
    lastTime = millis();  
  }  
}
```

Додаток Б
(Рекомендований)

КОПІЇ ПРЕЗЕНТАЦІЇ

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
КВАЛІФІКАЦІЙНА РОБОТА
на тему:

Інформаційна система збору та обробки даних в
сфері комунальних послуг

Виконав ст.гр. ІСТ 18-1
Керівник:

Кириченко А.А.
Бітченко О.М.

Харків 2024

Вступ

- В наш час врахування споживання гарячої та холодної води, електроенергії, теплоенергії в ЖКГ останнім часом стає актуальним завданням для всієї країни.
- На даний час це особливо актуальна задача, адже тисячі жителів України були вимушені покинути власні будинки, але сплачувати комунальні послуги зобов'язані, навіть якщо "платити, вже можливо, і не потрібно".

Створення системи обліку водопостачання дозволяє вирішити досить великий обсяг завдань, пов'язаних не лише з урахуванням, але й формування тарифних ставок. Постачальники отримують у максимально можливому обсязі гроші за надані ресурси, а споживачі, своєю чергу, платять лише за те, що споживали.



Моніторинг комунальних ресурсів

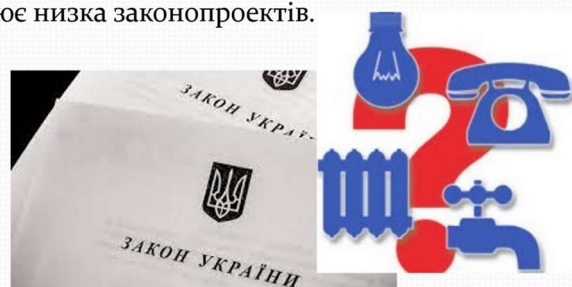
- Пристрій збору і передачі даних бере на себе функцію не тільки акумулювання і зберігання на деякому проміжку часу даних і подальшу їх передачу, а й виконання функцій обчислення значень спожитої води. Однак варто зазначити, що є два підходи - виконувати обчислення «на місці», або проводити обчислення на стороні сервера.



За рахунок автоматичної передачі комунальних показників вирішується проблема з оплатою комунальних послуг, адже на даний час цей процес виконується за участю споживача та спеціаліста, аби сформувати платежі, або ж через онлайн банкінг.

Проблеми в розвитку житлово-комунальної сфери

- У процесі дослідження нами з'ясовано проблеми в розвитку житлово-комунальної сфери. Відповідно, основними шляхами є удосконалення наступних складових економічної політики підприємства: цінової та тарифної політики; підвищення якості послуг; створення інституту відповідальних власників; політики соціального захисту; фінансово-стабілізаційної політики; системи управління; нормативно-законодавчої бази тощо. Регулювання взаємовідносин у сфері ЖКГ здійснюється на національному та регіональному рівні. Важливо знати, що галузь дослідження регулює низка законопроектів.



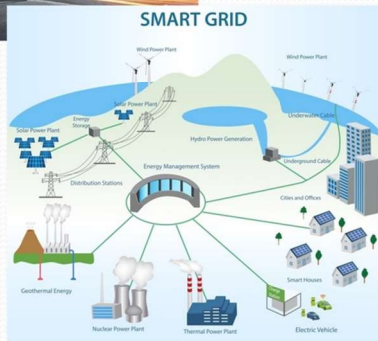
Огляд аналогових автоматизованих систем ЖКГ

- Система Tera Bills - це програмний комплекс, що виконує розрахунок нарахувань з оплати житлово-комунальних послуг, облік, накопичення та обробку інформації про надходження платежів квартиронаймачів, документування та відображення даних, а також формування різноманітних звітів.



ГЕРЦ створена з метою упорядкування розрахунків за енергоносії та забезпечення ефективного контролю за збором та проходженням платежів за житлово-комунальні послуги та для забезпечення можливості оплати за надані комунальні послуги без простоювання у чергах,

- Тож, на даний час дуже гострою є проблема із автоматизацією процесу збору комунальних показників від споживачів до підприємств, які надають саме комунальні послуги. На ринку починають з'являтися системи, які автоматично передають показники комунальних послуг напряму організаціям ЖКГ для подальшої їх обробки та нарахування виплат. За допомогою новітніх технологій, а саме таких як «розумні мережі» спроститься задача обліку показників комунальних послуг.



ІоТ ДЛЯ МОНІТОРИНГУ КОМУНАЛЬНИХ РЕСУРСІВ ЖКГ

- Цифровізація у сфері ЖКГ – новий вектор розвитку галузі, спрямований на автоматизацію процесів, створення зрозумілої системи нарахувань за комунальні послуги, а також збільшення ефективності роботи керівників підприємств і ресурсопостачальних організацій. Впровадження хмарних технологій у товариство власників житла є актуальним завданням. Звичайно, цей процес дуже не простий, проте він у результаті дозволяє здійснити чимало функцій, а також досягти нового ступеня управління багатоквартирним будинком з оперативним інформаційним обслуговуванням.



Arduino як ІоТ система

- Arduino – це апаратно-програмний комплекс для створення електронних пристроїв своїми руками, від маленького робота до системи «Розумний будинок».
- Arduino складається з апаратної і програмної частини. Тобто спочатку ви збираєте пристрій з «електронним мозком» на основі плати Arduino, а потім програмуєте отриманий модуль під свої завдання.



AWS

- Також не менш відомою є інформаційна системи AWS – це найпоширеніша у світі хмарна платформа з найширшими можливостями, що надає понад 200 повнофункціональних сервісів для центрів обробки даних по всій планеті. Багато компаній, що стали лідерами за швидкістю зростання, найбільшими корпораціями та передовими урядовими установами, використовують AWS для зниження витрат, підвищення гнучкості та прискореного впровадження інновацій.



Переваги:

- дозволяє обробляти великі масиви даних без втрати часу та енергозатрат;
- можливість шифрування даних;
- надійність та висока продуктивність;
- постійна доступність даних “тут і зараз”

АНАЛОГИ ЗАБОРУ ДАНИХ ЖКГ ПОКАЗНИКІВ.

Пристрій, який описується в даній кваліфікаційній роботі, суть якого полягає в систематизуванні та автоматизації забору даних з лічильників в житлово-комунальних господарствах. Дані, що збираються з лічильників надходять автоматично, щомісячно на сервер, після чого обробляються та аналізуються для формування подальших сплат за комунальні послуги, а саме: водопостачання. Така систематизація можлива за рахунок використання серверів, задля ефективній, швидкій та оперативній роботі, а саме AWS.



Вибір необхідних датчиків

Ватметр PeaseFair PZEM-022 – до функцій можна віднести: вимірювання електричних параметрів (напруги, ток); сигналізації перевантаження.

Датчик витрат води - Даний датчик ідеально підходить для автоматизації контролю за використанням води. Конструктивно датчик складається з пластикового корпусу з клапаном, водяного ротора та датчика Холла.

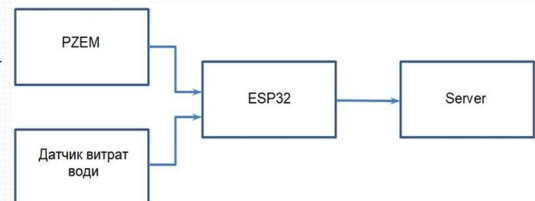
Сенсор потоку YF-B5 - витратомір на основі датчика Холла. Потік води розкручує внутрішній ротор, що приводить у рух датчик Холла, який видає відповідний імпульсний сигнал.

Ультразвуковий датчик витрати води Belimo FM0100F-SZ - датчик призначається вимірювання витрати рідких середовищ. Ідеально підходить для систем кондиціонування, опалення, вентиляції.



ПРИСТРІЙ ДЛЯ ЗБОРУ ПОКАЗНИКІВ КОМУНАЛЬНИХ ПОСЛУГ

- На рисунку наведено структурну схему розробки мого пристрою для збору показників води, для подальшої обробки отриманих даних.



Розглянемо детальніше блоки схеми розробленого пристрою:

- блок «PZEM» відповідає за фіксування води що споживає споживач та зберігає в енергонезалежній пам'яті;
- блок «ESP32» - плата DevKit з мікроконтролером ESP32, використовується для отримання витрат споживаної електроенергії за допомогою модулю PZEM та отримання показників витрат води за допомогою датчику витрат води, після чого отримані дані надсилаються до серверу для подальшої обробки та зберігання показників комунальних послуг користувача;
- блок «Датчик витрат води», використання даного датчику для отримання показників витрат води користувача;
- блок «Server» завантаження даних про спожиту воду до хманоного сховища та передачу їх на сервер.

- Зовнішній вигляд модуля PZEM наведено на рисунку

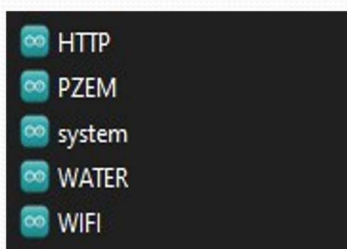


Основні характеристики:

- потужність: діапазон вимірювання 0 ~ 22 кВт;
- напруга: діапазон вимірювання 80 ~ 260 В змінного струму;
- струм: діапазон вимірювання 0 ~ 100А;
- робоча частота: 45-65 Гц;
- послідовний інтерфейс: UART;
- швидкість послідовного інтерфейсу: 9600;
- напруга живлення інтерфейсної частини: 5 В;
- точність вимірювання: 1% похибки.

Програмна частина

- Для реалізації програмної частини було використано середовище розробки Arduino IDE з додатковими налаштуваннями для роботи з ESP32 мікроконтролером.



Програма складається з основного файлу «system» та допоміжних файлів з описом функцій, перелік файлів наведено на рисунку.

В файлі «system» знаходиться код основної програми в якому підключаємо всі необхідні бібліотеки, виконуємо налаштування портів вводу/виводу та налаштування інтерфейсів зв'язку як приклад в нашому випадку UART інтерфейс для зв'язку з модулем PZEM.

Основний файл програми «system»



- Під час написання коду програми було використано наступні бібліотеки:
 - * - `#include <WiFi.h>` – дану бібліотеку використовуємо для роботи з бездротовою мережі Wi-Fi, що забезпечує нам канал передачі даних до серверу;
 - * - `#include <ArduinoJson.h>` – використовуємо для роботи з JSON форматом даних що отримуємо від серверу для встановлення початкових показників електроенергії та води, також використовуємо даний формат для передачі даних до серверу;
 - * - `#include <HttpClient.h>` – використовуємо для використання HTTP протоколу для відправки та отримання даних до серверу;
 - * - `#include <PZEM004Tv30.h>` – використовуємо для роботи з модулем PZEM для отримання даних показників використаної електроенергії споживачем.

The screenshot shows a web browser window with the URL localhost:8080. The page title is 'Kyrychenko Alina'. The interface includes a navigation bar with 'All clients', 'Add client', and 'Edit client'. The main content area displays two tables:

Example all clients

Address	Number apartment	Energy	Water
Nauky Ave. 14	313	175	200
Nauky Ave. 14	311	215	57
Nauky Ave. 14	309	330	100

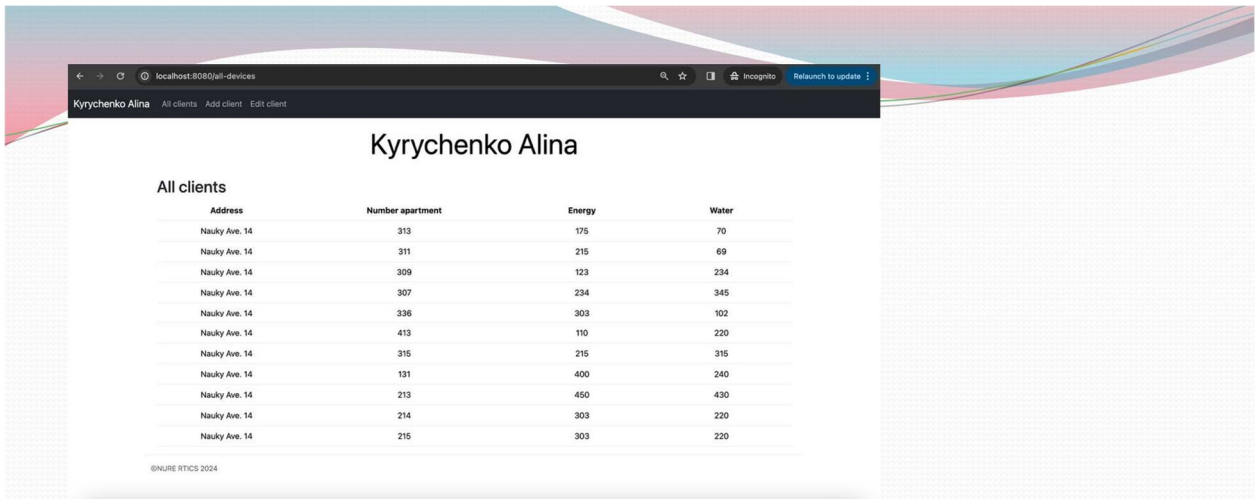
Example edit clients

Address	Number apartment	Energy	Water	Update	Delete
Nauky Ave. 14	313	175	200	Update	Delete
Nauky Ave. 14	311	215	57	Update	Delete
Nauky Ave. 14	309	330	100	Update	Delete

©NUIRE RTICS 2024

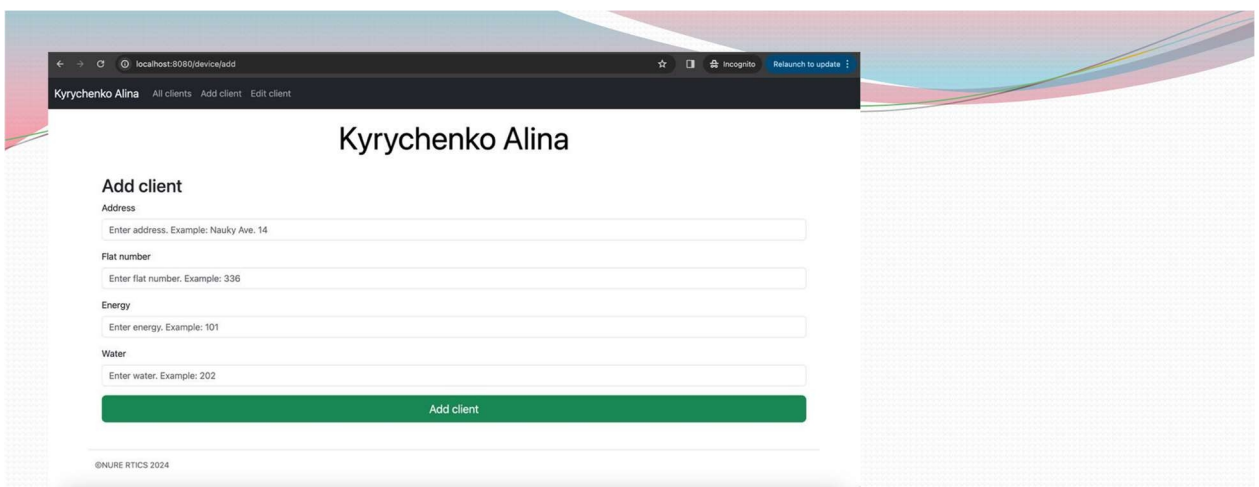
Контролер WebController оброблює запити з веб браузеру після чого в залежності від функції обробки ми відображаємо веб сторінку в браузері.

На рисунку 4.12 приведено відображення головної сторінки за запитом (відобразити адресу, номер квартири, споживання електроенергії та води).

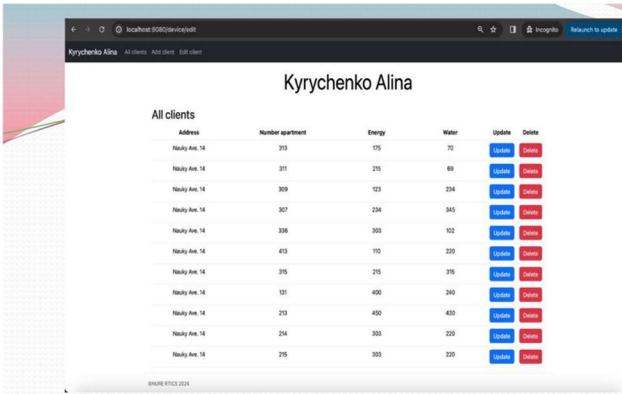


На головній сторінці приведено приклад відображення всіх клієнтів та сторінка редагування.

На рисунку вище приведено сторінку відображення всіх користувачів з поточними показниками комунальних послуг, адреса сторінки «/all-devices».

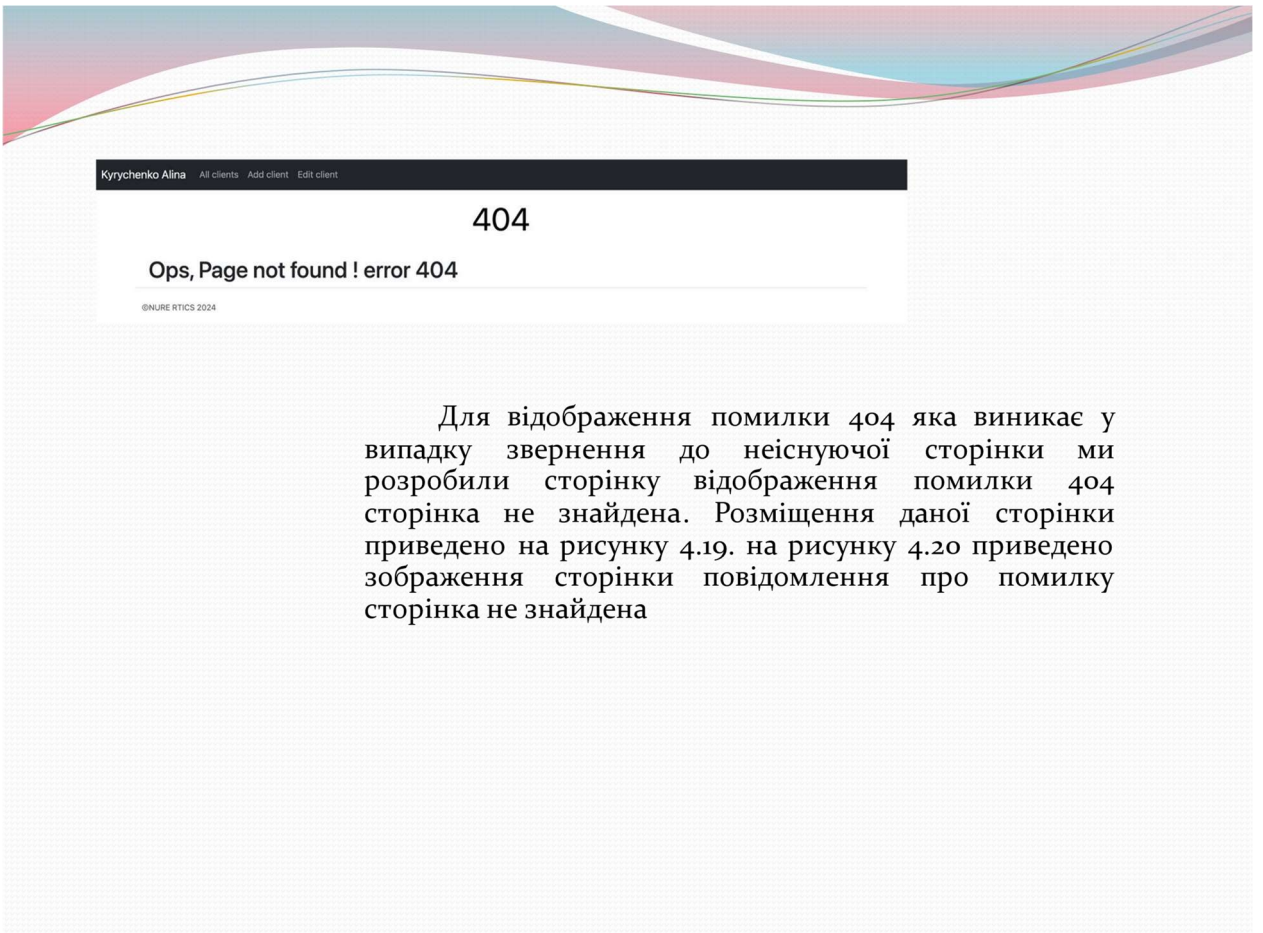
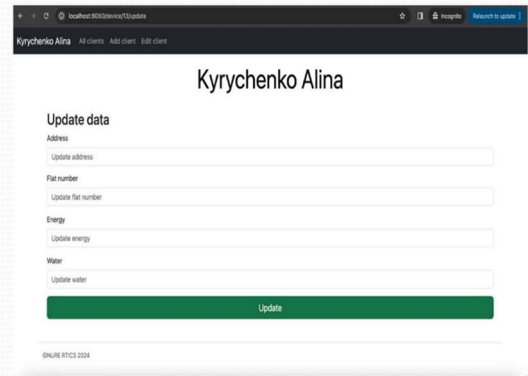


На рисунку вище приведено відображення сторінки додати клієнта за посиланням /device/add, де вказуємо адресу користувача, номер квартири, поточні показники лічильника електроенергії, поточні показники лічильника води.



На рисунку зліва наведено відображення сторінки редагування запису клієнта, за посиланням /device/edit. Розроблено два режими редагування запису користувача це оновлення даних або видалення запису користувача.

У випадку кліку на кнопку «update» ми переходимо до сторінки оновлення записів користувача за посиланням /device/{id}/update, де ми можемо від коректувати всі поля користувача. Сторінка оновлення приведена на рисунку праворуч.



Для відображення помилки 404 яка виникає у випадку звернення до неіснуючої сторінки ми розробили сторінку відображення помилки 404 сторінка не знайдена. Розміщення даної сторінки приведено на рисунку 4.19. на рисунку 4.20 приведено зображення сторінки повідомлення про помилку сторінка не знайдена

Висновки

- Метою магістерської кваліфікаційної роботи була розробка автоматичної системи збору даних комунальних послуг з розрахунком вартості за місяць. В першому розділі роботи було розглянуто принципи побудови комерційного обліку та деякі аспекти впровадження автоматизованих систем. В другому розділі роботи проведено огляд та аналіз аналогічних систем таких як система Tera Bills, автоматизована система управління підприємствами житлово-комунального господарства (АСУП ЖКГ) тощо. При схемотехнічній розробці системи було проведено обґрунтування вибору мікроконтролера та обрано необхідні датчики. В четвертому, програмному, розділі наведено програмні коди підключення датчиків до мікроконтролера та розроблено Web додаток.



Дякую за увагу!



Додаток В
(Обов'язковий)

ВІДОМІСТЬ АТЕСТАЦІЙНОГО ПРОЕКТУ

