Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)
Факультет інформаційних технологій
(факультет)
Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра
(назва освітньо-кваліфікаційного рівня)

студента *Закликоцького Богдана Андрійовича*
(ПІБ)

академічної групи *122-18-1*
(шифр)

спеціальності *122 Комп'ютерні науки*
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*
(назва освітньої програми)

на тему: *Розробка автоматизованої інформаційної системи "розумний дім" яка заснована на мікропроцесорі Arduino*

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	*доц. Кабак Л.В.*			
розділів:				
спеціальний	*доц. Кабак Л.В.*			
економічний	*доц. Касьяненко Л.В.*			
Рецензент	*доц. Шедловський І.А.*			
Нормоконтролер	*доц. Гуліна І.Г.*			

Дніпро
2022

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
завідувач кафедри
програмного забезпечення комп'ютерних систем
(повна назва)

_____ І.М. Удовик
(підпис) (прізвище, ініціали)

« » _____ 2022 року

ЗАВДАННЯ
на кваліфікаційну роботу

бакалавра
(назва освітньо-кваліфікаційного рівня)

студента *122-18-1* *Закликоцького Б.А.*
(група) (прізвище та ініціали)

тема кваліфікаційної роботи *Розробка автоматизованої інформаційної системи "розумній дім" яка заснована на мікропроцесорі Arduino*

затверджена наказом ректора НТУ «ДП» від « » .2022 р. №

Розділ	Зміст виконання	Термін виконання
Спеціальний	***На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми***	***13.05.2022 р.***
Економічний	***Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки***	***27.05.2022 р.***

Завдання видав _____ *доц. Кабак Л.В.*
(підпис) (посада, прізвище, ініціали)

Завдання прийняв до виконання _____ *Закликоцький Б.А.*
(підпис) (прізвище, ініціали)

Дата видачі завдання: *15.10.2022 р.*

Термін подання кваліфікаційної роботи до ЕК: *11.06.2022 р.*

РЕФЕРАТ

Пояснювальна записка: __ с., __рис., __табл.,__дод.,___джерел.

Темою кваліфікаційної роботи є «Розробка автоматизованої інформаційної системи “розумній дім” яка заснована на мікропроцесорі Arduino»

Мета кваліфікаційної роботи: створення інформаційної системи, що надаватиме можливість користувачу за допомогою мобільного пристрою на базі операційної системи Android:

1. Керувати освітленням і температурою в домі
2. Моніторити стан освітлювальних пристроїв і температуру в домі
3. Створювати або видаляти сценарії користувача за шаблоном

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточняється постановка завдання.

У першому розділі проводиться дослідження предметної області та існуючих рішень, визначається актуальність завдання та призначення розробки, розроблюється постановка завдання.

У другому розділі обирається платформа для розробки, виконується проектування програми і її розробка, наводиться опис алгоритму і структури функціонування системи, визначаються вхідні і вихідні дані, наводяться характеристики складу параметрів технічних засобів, описується робота програми.

В економічному розділі визначається трудомісткість розробленого програмного продукту, проводиться підрахунок вартості роботи по створенню застосунку та розраховується час на його створення.

Практичне завдання полягає у розробці інформаційної системи, що дозволяє користувачам за допомогою мережі інтернет з будь-якої точки земного шару керувати системою «Розумний дім», а саме: керувати освітленням і температурою в домі, моніторити стан освітлювальних пристроїв і температури в будинку, створювати сценарії користувача за шаблоном наведеним в додатку, видаляти сценарії створені користувачем або іншими користувачами які підключені до додатку і в яких є доступ до мережі інтернет. Сценарії дозволяють без втручання користувача автоматично по обраним категоріям робити ті чи інші дії з приладами підключеними до системи. Зберігання на себе бере база даних(БД), а клієнтський додаток або контролер розумного будинку звертається до БД за допомогою POST-запитів до PHP веб сторінки для отримання або відправлення даних з/до БД.

Актуальність програмного додатку визначається великим ростом популярності інтернету речей, адже інформаційно система «Розумний дім» дає змогу керувати процесами, що відбуваються в будинку з будь-якого місця. Наприклад Ви не знаєте чи вимкнули світло в спальній кімнаті коли вийшли з будинку, а з інформаційною системою «Розумний дім» Ви будете напевно знати що, де і коли відбувається в вашому будинку через мережу інтернет.

Список ключових слів: ДОДОТОК, ІНФОРМАЦІЙНА СИСТЕМА, МОНІТОРИНГ, КЛІЕНТ, КОНТРОЛЛЕР, БАЗА ДАНИХ, ВЕБ СТОРІНКА

ABSTRACT

Explanatory note: __p., __figs., __ appx., __ sources.

The theme of the qualification work is "Development of an automated smart home information system based on the Arduino microprocessor"

The purpose of the qualification work: to create an information system that will enable the user with a mobile device based on the Android operating system:

1. Control the lighting and temperature in the house
2. Monitor the condition of lighting devices and temperature in the house
3. Create or delete user scripts based on a template

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the field of its application, provides a justification for the relevance of the topic and clarifies the task.

In the first section the research of the subject area and existing decisions is carried out, the urgency of the task and the purpose of development is determined, the statement of the task is developed.

The second section selects the platform for development, performs program design and development, describes the algorithm and structure of the system, determines the input and output data, provides characteristics of the parameters of technical means, describes the program.

The economic section determines the complexity of the developed software product, calculates the cost of work to create an application and calculates the time to create it.

The practical task is to develop an information system that allows users to use the Internet from anywhere in the world to control the system "Smart Home", namely: control lighting and temperature in the house, monitor the status of lighting and temperature in the house, create scenarios user according to the template provided in the application, delete scripts created by the user or other users who are connected to the application and who have access to the Internet. Scripts allow you to automatically perform certain actions on the selected categories with the devices connected to the system without user intervention. Storage is assumed by the database (DB), and the client application or smart home controller accesses the database via POST-requests to the PHP web page to receive or send data from / to the database.

The relevance of the software application is determined by the great growth of the Internet of Things, because the information system "Smart Home" allows you to control the processes taking place in the house from anywhere. For example, you do not know whether you turned off the light in the bedroom when you left the house, and with the information system "Smart Home" you will know for sure what, where and when is happening in your home via the Internet.

List of keywords: APPENDIX, INFORMATION SYSTEM, MONITORING, CLIENT, CONTROLLER, DATABASE, WEB PAGE

ЗМІСТ

РЕФЕРАТ 3
ABSTRACT 4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ 7
ВСТУП 8
РОЗДІЛ 1 10
АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ 10
1.1. Загальні відомості з предметної галузі 10
1.2. Призначення розробки та галузь застосування 13
1.3. Підстави для розробки 15
1.5. Вимоги до програми або програмного виробу 16
1.5.1 Вимоги до функціональних характеристик 16
1.5.2. Вимоги до інформаційної безпеки 18
1.5.3. Вимоги до складу та параметрів технічних засобів 18
1.5.4. Вимоги до інформаційної та програмної сумісності 19
РОЗДІЛ 2 20
ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ 20
2.1. Функціональне призначення програми 20
2.2. Опис застосованих математичних методів 21
2.3. Опис використаних технологій та мов програмування 21
2.4. Опис структури системи та алгоритмів її функціонування. 25
2.5. Обґрунтування та організація вхідних та вихідних даних програми 42
2.6. Опис розробленої системи 45
2.6.1 Використані технічні засоби. 45
2.6.2 Використані програмні засоби. 46
2.6.3 Виклик та завантаження програми. 47
2.6.4. Опис інтерфейсу користувача. 48
Розділ 3 58
Економічний розділ 58
3.1. Визначення трудомісткості розробки програмного забезпечення 58
3.2. Розрахунок витрат на створення програми 61

ВИСНОВКИ 63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 65
ДОДАТОК А 67
ДОДАТОК Б 68
ДОДАТОК В 69
КОД ПРОГРАМИ 69
ДОДАТОК Г 82
Відзив керівника економічного розділу 82
ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ 83

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

APK – Android Package (формат архівних файлів-додатків для «Android»);

SDK – Software Development Kit (набір із засобів розробки);

PHP – Hypertext Preprocessor (гіпертекстовий препроцесор);

ПЗ – програмне забезпечення;

БД – база даних;

ІС – інформаційна система;

ПЗ – програмне забезпечення;

ОС – операційна система;

ІТ – інформаційні технології.

АВР – автоматичне введення резерву

ВСТУП

Поняття «розумний дім» у багатьох асоціюється з будинком який готує їжу, прибирає сам себе, підготовлює комфортну температуру і рівень світла під кожного члену сім'ї.

Розумний будинок — це технологія, що дозволяє управляти електронними елементами будинку без застосування фізичного участі самої людини. Основна суть — це управління предметами з мінімальними фізичними витратами.

Метою даної кваліфікаційної роботи є вивчення інструментальних засобів Android Studio, Arduino IDE, PHP та роботи з БД. Визначення і вивчення різних методів обміну даними між сервером і клієнтом, а саме приймання даних на стороні серверу їх сортування по таблицям в БД, та створення пакету даних для відправки на клієнтську сторону і подальшим аналізом прийнятого пакету даних. Дослідження як найкрашчого способу для розробки програми на базі ОС Android для якнайшвидшої роботи, написання і оптимізація програмного коду для мікроконтролера ATMEGA 2560 на базі плати Arduino MEGA 2560 для якнайшвидшої роботи одно процесорного контролеру і швидкого обміну даними з сервером.

Чому розумний дім – майбутнє? За останнє десятиліття ми всі звикли керувати аспектами нашого життя за допомогою технологій, від онлайн-банкінгу до інтернет-магазинів, Інтернет зробив життя набагато зручнішим. То чому б такій інтелектуальній революції не відбутися у нас вдома

Що таке технологія розумного дому? Розумний дім — це інформаційна система в якій кожен аспект можна керувати за допомогою цифрового пристрою. Вхідні двері, дверний дзвінок, будь-яке освітлення, опалення в різних кімнатах будинку, температура гарячої води, навіть шторами можна керувати дистанційно.

Які переваги розумного будинку? Розумний будинок робить перебування вдома набагато комфортнішим, а також може дати вам спокій, що все працює правильно, навіть імітувати вашу присутність поки ви відпочиваєте на курорті

Розумна технологія спрощує роботу по будинку, зменшуючи потребу в багаточисельних проводах і пультах дистанційного керування для інших пристроїв.

Деякі компанії які розробляють систему розумного будинку зосереджуються на безпеці – інтегрована система спостереження розумного будинку може дозволити вам стежити за тим, що відбувається, де б ви не були. Ви також можете придбати системи, які захищають ваш будинок від пожежі та затоплення. Розумний дім дозволяє краще контролювати і знати більше про те, що відбувається у вашому будинку.

Як працює розумний дім? Технологія розумного дому може бути представлена у вашому домі частинами, відповідно до ваших пріоритетів. Однак інформаційна система «Розумний будинок» зосереджена на інтегрованій технології, яка з'єднує різні елементи вашого будинку в одному центрі – вашому смартфоні.

Ви також можете використовувати ІС, щоб дізнатися про стан вашої оселі через розумний додаток, навіть якщо ви не вдома, ви точно знаєте, що відбувається. Ви також можете використовувати цю систему, щоб вимкнути опалення, як тільки система попередить вас про проблему - навіть якщо вас немає вдома. Такі системи є останньою конструкцією розумного будинку, яка без зусиль допомагає власникам будинків, де б вони не були.

Модель такої системи запропоновано у дипломному проекті.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1.Загальні відомості з предметної галузі

Розумний будинок (англ. Smart House) - сучасна житлова будівля, організована для полегшення використання високотехнологічного обладнання. Електронні пристрої в розумному домі можна об’єднати в домашню plug and play – мережу з доступом до загальнодоступних мереж. Концепція «розумного будинку» була запропонована Інститутом інтелектуального будівництва у Вашингтоні в 1970-х роках: «Будівля, яка ефективно використовує робочий простір»

Варто розділяти поняття «розумний дім» та «системи життєзабезпечення». Окремі системи мають лише необхідні інтерфейси управління і мониторингу. Концепція «Системи інтелектуального управління будинком» передбачає новий підхід в організації життєзабезпечення будівлі, при якому за рахунок кластеру програмно-апаратних засобів значно зростає ефективність функціонування та надійність управління всіх структур експлуатації та виконавчих пристроїв будівлі.

Головною особливістю розумних будівель є інтеграція різноманітних підсистем різних виробників в єдиний комплекс управління. Під «розумною будівлею» слід розуміти систему, яка повинна вміти розпізнавати конкретні ситуації, що відбуваються в будівлі, і відповідним чином реагувати: одна з систем може керувати поведінкою інших систем за розробленими алгоритмами. Англійське слово intelligent, що буквально означає «розумний», у поєднанні зі словом building означає «гнучкий і адаптований».

Будівля спроектована таким чином, що всі системи управління можна інтегрувати одна з одною з мінімальними витратами, а їх обслуговування буде організовано оптимально.

Згодом будівлі знайдуть «штучний інтелект». Тоді повною мірою можна буде називати їх інтелектуальними. Системи зможуть відслідковувати роботу і

стан всієї «начинки» будівлі, включаючи огороджувальні конструкції, і самостійно приймати рішення в обставинах, що змінюються.

Під терміном «розумний дім» зазвичай розуміють інтеграцію в єдину систему керування будівлею наступних систем:

- Систему опалення, вентиляції та кондиціювання
- Охоронно-пожежну сигналізацію, систему контролю доступу в приміщення, контроль протікання води, витоку газу
- Систему відеоспостереження
- Мережі зв'язку (у тому числі телефон та локальна мережа будівлі)
- Систему освітлення
- Систему електроживлення будівлі (АВР, дизель-генератори)
- Механізацію будівлі (відкриття/закриття воріт, шлагбаумів, електропідігрівану систему сходів тощо)
- Управління з одного місця аудіо-, відеотехнікою, домашнім кінотеатром, мультирум
- Телеметрія - віддалене стеження за системами
- IP-моніторинг об'єкта - віддалене управління системами по мережі
- GSM-моніторинг — віддалене інформування про інциденти в будинку та управління системами будинку через телефон.

Сьогодні технології дозволяють створювати компоненти домашньої автоматизації один за одним, вибираючи функції розумного дому, які вам дійсно потрібні. Модульна структура дозволяє створювати недорогі системи і гарантує 100% використання.

Один із найстаріших і найненависніших проектів – котедж Білла Гейтса. Цей проект, як і багато науково-фантастичної літератури, викликав незліченну кількість міфів про «розумний дім».

Функція вимкнути все світло однією кнопкою, як і змога вмикати та вимикати його з різних місць, зараз активно реалізується чи не у всіх проектах домашньої автоматизації.

У Європі проекти автоматизації приватних будинків і квартир готує персонально розробник і виробник систем, інсталятору ж відводиться вага фактично стандартних, але кваліфікованих монтажників, які працюють за схемою.

Існує дві різні концепції побудови подібних систем: централізоване (наприклад, IHC від Lexel) та децентралізоване на основі інсталяційної шини (EIB, LonWork, Crastron та ін.)

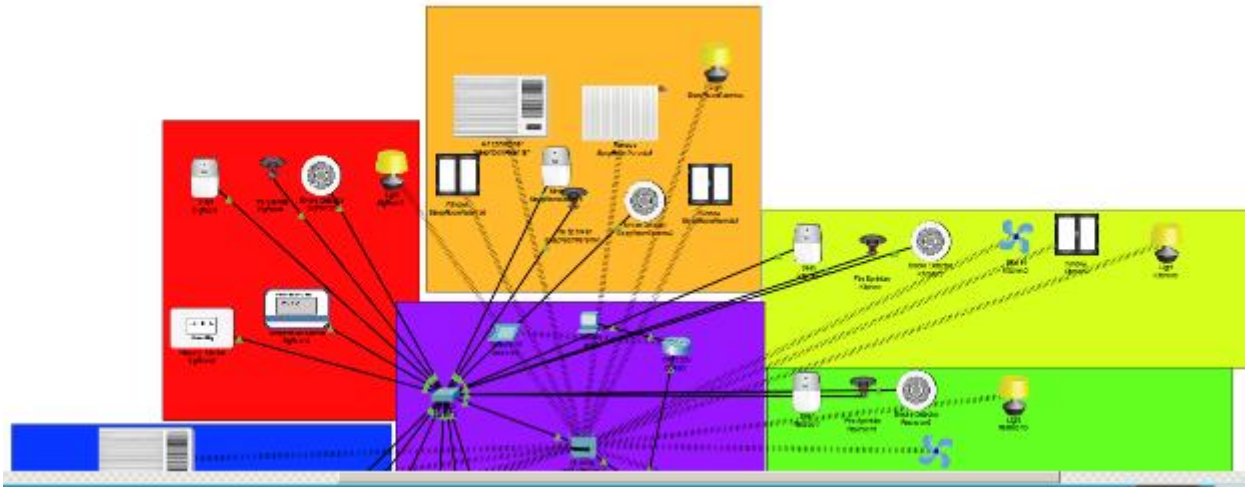


Рис. 1.1– Спрощена схема розумного будинку

Функції розумного будинку включають:

Управління мікрокліматом

Система постійно вимірює температуру в кожній кімнаті окремо і підтримує її на заданому вами рівні, безпосередньо керуючи клапаном радіатора або заслінкою кондиціонера, а також автоматично вмикаючи або вимикаючи вентиляцію, коли це необхідно. Система допомагає вам щодня економити гроші за допомогою різних режимів роботи системи. Режим змінюється відповідно до розкладу або команди користувача. Для кожного режиму достатньо лише один раз встановити температуру на сенсорному дисплеї в кімнаті. Якщо вікна в

кімнаті відкриті для провітрювання, система опалення/кондиціонування автоматично вимкнеться для економії енергії.

Контроль освітлення

Елементи керування освітленням дозволяють користувачам створювати сцени освітлення з необмеженої кількості джерел світла різної яскравості, вмикаючи їх одночасно або із затримкою, імітуючи такі ефекти, як «б'ючи вогні». Використовуючи спеціальний диммер, ви можете не тільки змінити яскравість ввімкнення світла, а й скільки часу потрібно для досягнення цієї яскравості. Можливість постійного керування освітленням в основному використовується в офісі і може підтримувати освітлення робочої поверхні, встановлене користувачем, незалежно від того, закрите хмарами сонце чи небо. Автоматична активація зовнішніх повідомлень за часом доби та присутності людей не тільки забезпечить додатковий комфорт, але й відлякує непроханих гостей.

Ефект присутності

Без користувачів будинок може імітувати звичний спосіб життя власника, включаючи освітлення та музику вночі, створюючи ефект присутності.

1.2.Призначення розробки та галузь застосування

Розробка автоматизованої інформаційної системи "розумній дім" яка заснована на мікропроцесорі Arduino.

Для створення такої інформаційної системи використовується як і програмні засоби збору і обробки інформації так і апаратні (рис 1.1).

До апаратних засобів збору інформації про будівлю можна віднести мікроконтролер з периферією, який протягом певного періоду часу збирає інформацію з периферійних пристроїв та формує пакет даних для подальшої відправки пакету на сервер, інтернет контролер, який дасть змогу зв'язатися з сервером. Сервер на який поступають і зберігаються данні. Клієнтський пристрій

на базі Android 7.0 або вище для віддаленого перегляду стану будинку і здійснення управління периферією контролера.

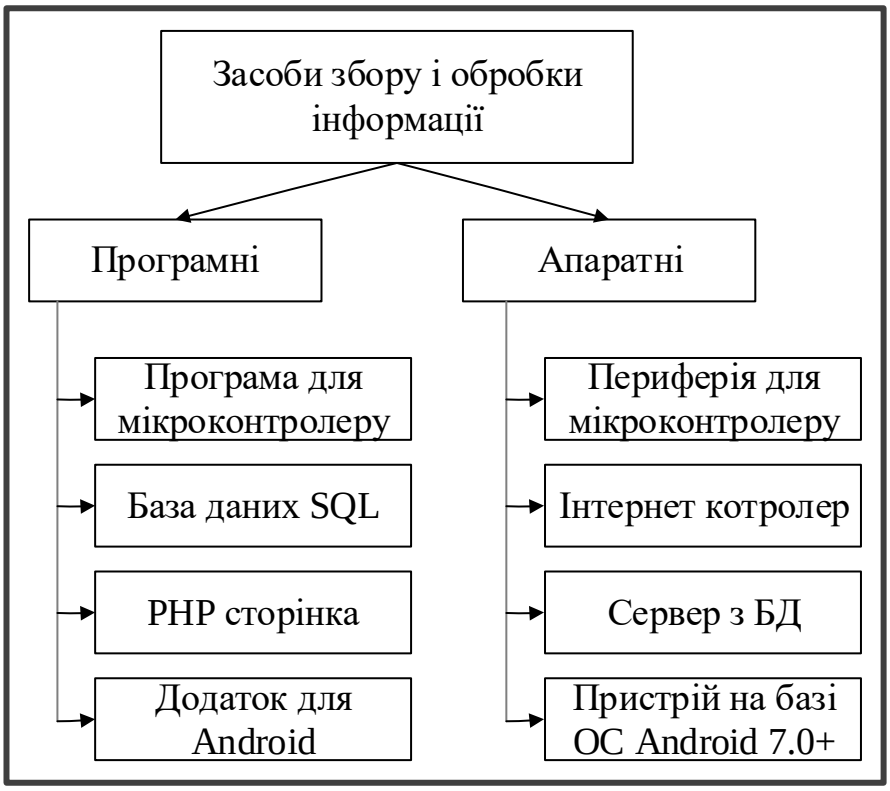


Рис 1.2 Засоби збору і обробки інформації

До програмних засобів відноситься: по-перше мікроконтролер з програмою, яка зможе зібрати інформацію з периферії, обробити та «запакувати» дані для відправки на сервер. По-друге сервер з підтримкою баз даних SQL, та підтримкою мови програмування PHP. По-третє мобільний додаток написаний на мові JAVA для пристроїв Android на базі ОС Android 7.0 або вище, написаний з використанням бібліотеки Volley яка дає змогу відправляти HTTPS запити на сервер.

Інформаційну систему можна поліпшити використовуючи більш дорогий аналог мікропроцесору з декількома ядрами наприклад Raspberry PI 3 B+, так як незначним недоліком є однопоточність процесора, що не дає змогу одночасно приймати данні з сервера та виконувати зчитування з периферії, мікропроцесор зробить це послідовно, тобто користувач інформаційної системи не зможе увімкнути або вимкнути елемент периферії поки йде оновлення даних с сервером на контролері. Вважаю цей недолік незначним для створення інформаційної

системи, так як він не здійсню вплив для інших елементів ІС, а лише для користувача і тільки в тому випадку коли користувач буде знаходитися в ситуації коли контролер почне автоматичне оновлення даних і в цей же момент користувач увімкне або вимкне елемент периферії на контролері.

1.3.Підстави для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу.

Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки (виконання кваліфікаційної роботи) є:

- освітня програма 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» №___від __.___.2022 р;

завдання на кваліфікаційну роботу на тему «Розробка автоматизованої інформаційної системи "розумній дім" яка заснована на мікропроцесорі Arduino»

1.4. Постановка завдання

Завданням даної роботи є розробка автоматизованої інформаційної системи "розумній дім" яка заснована на мікропроцесорі Arduino.

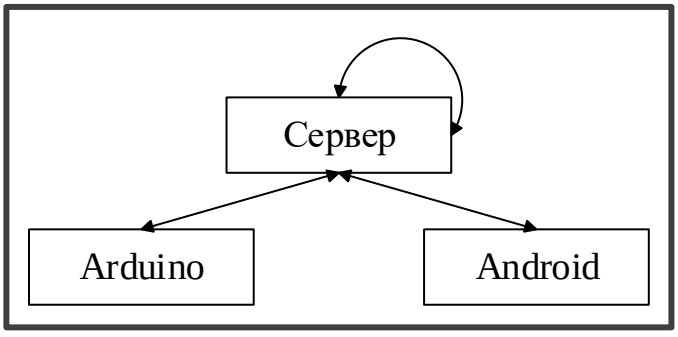


Рис 1.3 Загальна схема роботи інформаційної системи

В результаті необхідно спроектовати та розробити програму для мікропроцесору, серверну частину з PHP програмами та SQL базами даних, та клієнтський Android додаток який буде виступати в ролі панелі оператора і з якого можна задавати сценарії роботи елементів периферії контролеру.

Кінцевий продукт матиме дві лінії зв’язку з інформаційною системою:

– апаратний;

– програмний;

До апаратної лінії зв’язку відноситься безпосередньо взаємодія з елементами периферії мікроконтролера, а до програмної Android додаток з відображенням всіх елементів периферії в реальному часі та змогою взаємодії з ними.

1.5.Вимоги до програми або програмного виробу

1.5.1 Вимоги до функціональних характеристик

Кінцевий продукт матиме такі елементи:

– контролер на базі мікропроцесора Arduino;

– сервер

– мобільний додаток на базі ОС Android 7.0 або вище;

Програма для контролера на базі мікропроцесора Arduino має виконувати такі функції:

– зчитування даних з периферії;

– зберігання інформації про периферію;

– зміна цільового стану елементів периферії з аналогового вводу за допомогою кнопок;

– зміна цільового стану елементів периферії по отриманих даним зі серверу;

– обробка і розподілення отриманих даних по змінним в програмі;

– відправка оброблених даних на сервер;

– отримання даних про периферію з сервера;

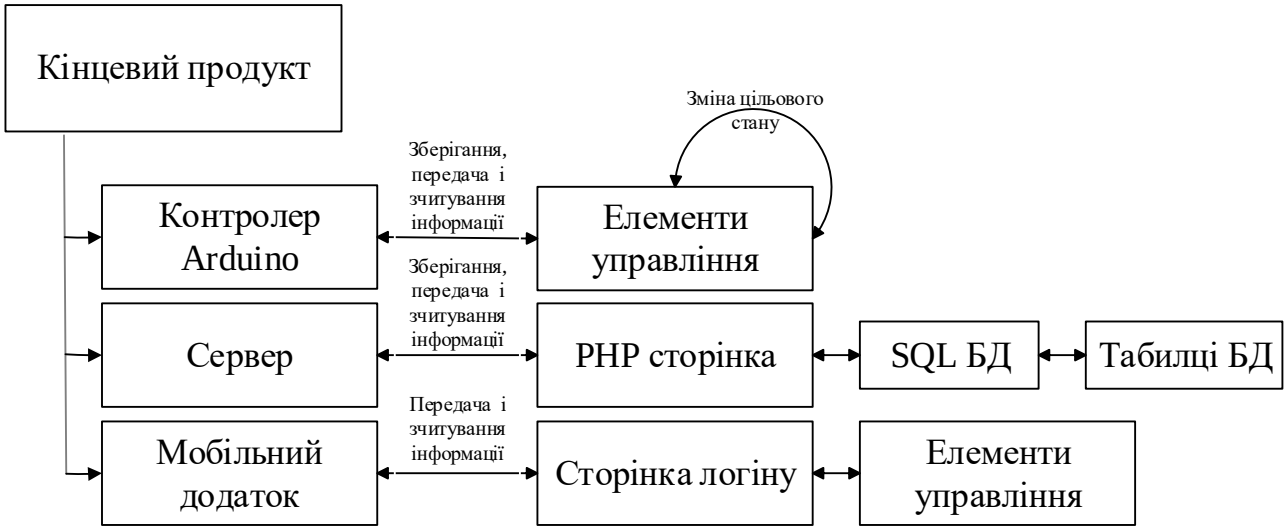


Рис 1.4 Спрощена схема кінценвого прокукту

Сервер повинен містити:

– SQL базу даних з таблицями розподілених по категоріям відповідно до різних типів периферії контролеру;
– таблиця де зберігається інформація про стан увімкнутий чи вимкнутий той ти інший елемент периферії;
– таблиця де зберігаються дані про поточну і цільову температуру в різних частинах будинку;
– таблиця з описом сценарію роботи елементів периферії контролеру;

Мобільний додаток на базі ОС Android 7.0 або вище повинен містити наступні функції:

– авторизування користувача;
– відображення стану освітлювальних приладів;
– відображення поточної температури різних частин будинку;
– можливість встановлення цільової температури для різних частин будинку;
– відображення встановлених сценаріїв роботи;
– створення сценаріїв роботи по шаблону;
– видалення сценаріїв;

1.5.2. Вимоги до інформаційної безпеки

Основні вимоги до інформаційної безпеки:

– цільність даних;

– конфіденційність інформації;

– впровадження політики доступу;

– доступність інформації;

Тільки авторизований користувач зможе переглядати інформацію в мобільному додатку.

Використання протоколу обміну HTTPS забезпечує шифрування всіх основних елементів запиту. В свою чергу використання POST запиту який створений під контрений PHP код сторінки серверу ускладнює прочитання і розуміння переданої інформації для нецільового користувача.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для роботи мобільного додатку необхідна платформа на базі операційної системи ОС Android 7.0 або вище з доступом в інтернет.

Програма для контролеру розроблена спеціально під мікроконтролер на базі Arduino MEGA 2560 з використанням веб-сервер шилд Ethernet Shield W5100 Arduino.

Для коректної роботи серверу потрібно:

– від 150 гігабайт SSD;

– ЦП: Pentium 4;

– 512 мегабайт відеопам’яті;

– 4 гігабайти оперативної пам’яті;

– доступ в інтернет на швидкості від 10 MB/s;

1.5.4. Вимоги до інформаційної та програмної сумісності

Інформаційна система “розумній дім” розроблена спеціально для мікроконтролера Arduino MEGA 2560 з використанням веб-сервер шилд Ethernet Shield W5100 Arduino. Програма написана на модифікованій мові програмування C++ спеціально для даних видів контролерів, програмі для розкриття повного потенціалу потрібен доступ до мережі інтернет. В програмі передбачено всі можливі варіанти поводження користувача. Для мобільного додатку використовувалась мова програмування JAVA з використанням бібліотеки Volley, при цьому для роботи програми потрібна операційна система ОС Andriod 7.0 або вище з постійним доступом в інтернет.

Для серверу:

– можливість створити базу даних;

– підтримка PHP коду;

РОЗДІЛ 2
ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення програми

Результатом виконання кваліфікаційної роботи повинена бути розроблена інформаційна система з її функціональними елементами:

– контролер на базі мікропроцесора Arduino;

– сервер;

– мобільний додаток на базі ОС Android 7.0 або вище;

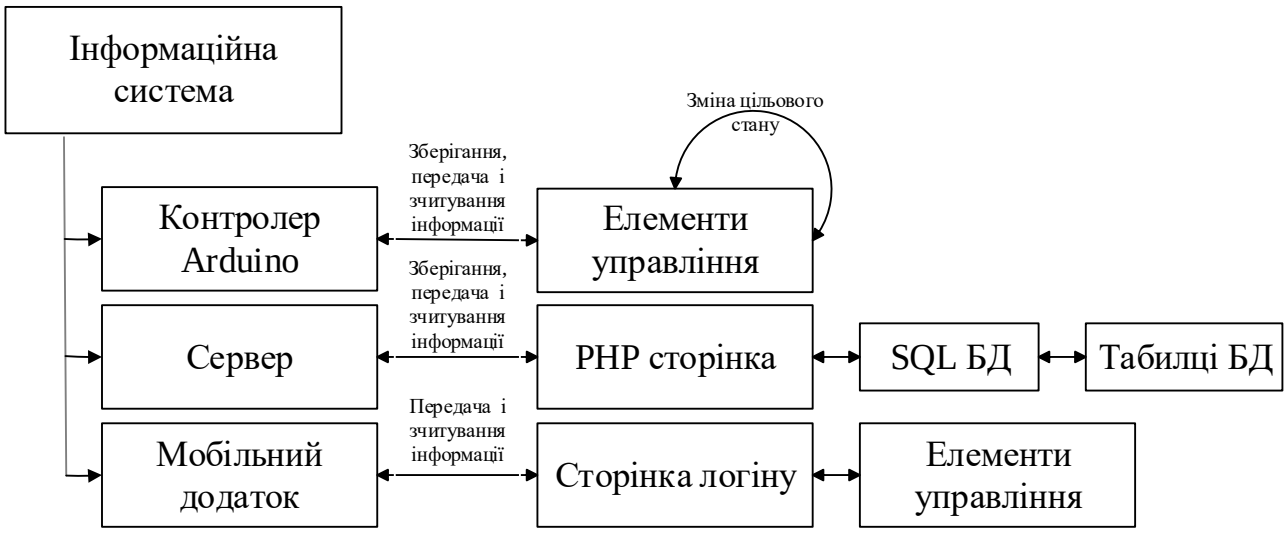


Рис 2.1 Спрощена схема кінценвого прокукту

Розроблена інформаційна система придназначена для управління оселею з мінімальними фізичними витратами. Елементи інформаційної системи будуть комунікуватися за допомогою мережі інтернет між собою. Причому зв’язок між всіма її елементами проходить через сервер, в якому за допомогою бази даних і зберігається вся інформація щодо стану елементів розумного будинку.

Користувач за допомогою телефону зможе з будь якої точки світу увімкнути або вимкнути байдуже який елемент розумного будинку, або встановити температуру обігріваючого пристрою. Це дає змогу економити на електроенергії в той час коли нікого немає вдома, встановивши меншу температуру опалення, а повертатися вже в теплий будинок заздалегідь встановивши сценарій роботи або в ручному режимі.

Основне призначення — це спрощення взаємодії між користувачем і його будинком.

Основний функціонал мобільного додатка полягає в дистанційному керуванню контролеру через мережу інтернет. Додаток в собі містить зручні елементи управління розміщені в відсортованому вигляді.

Основний функціонал серверу полягає в збереженні інформації про стан оселі і дає змогу як контролеру отримати актуальну інформацію про те в який стан треба встановити той чи інший елемент будинку, і отримати мобільному додатку актуальну інформацію про стан будинку.

Основний функціонал контролеру Arduino полягає в підтриманні встановленого режиму освітлення або температури в оселі, передачі архівованих даних на сервер.

2.2. Опис застосованих математичних методів

Під час проектування та розробки даної інформаційної системи використовувалися лише прості арифметичні дії. Математичні методи не використовувалися.

2.3. Опис використаних технологій та мов програмування

Розібравши різні системи розумного будинку можна виявити, що більшість систем базуються на дуже схожій ідеології, і відрізняються тільки в незначних деталях. Окрім цього в них присутні однакові основні деталі. Це не дивно тому, що весь наш побут, не залежно від місця проживання або фінансового стану, ідентичний. У кожного вдома стоїть обігріваючий елемент, є розетки, вікна, двері і освітлювальні прилади. Тому незалежно від виробника інформаційної системи всі вони націлені на роботу з саме такими приладами, в своєму мінімумі.

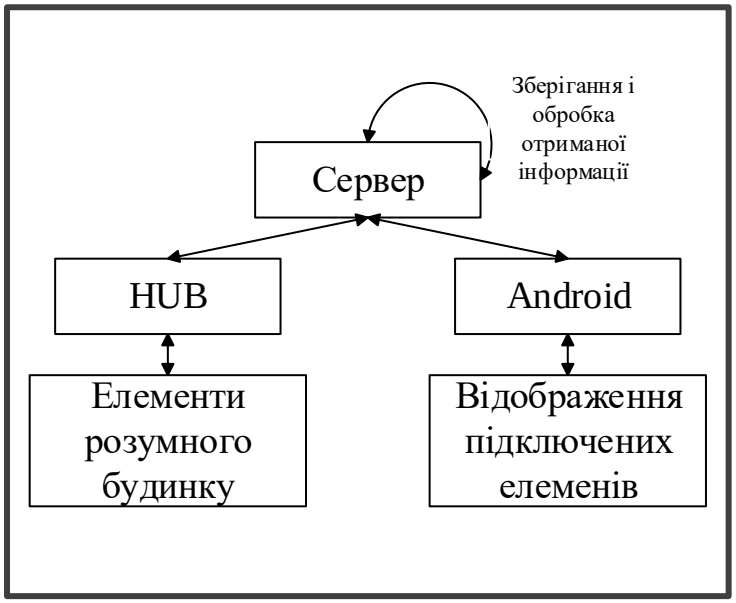


Рис 2.2 Типова схема розумного будинку з концентраційним HUB

Ця схожість основного функціоналу серед різних систем дозволяє розділити інформаційну систему на три складові, кожна складова підтримує роботу системи в штатному режимі. Перша з них це контролер (HUB), він дозволяє підключити всі елементи будинку до інформаційної системи і подальшого управління елементами, також контролер зв'язується з сервером для віддаленого контролю підключеними елементами. Друга з складових це сервер, він оброблює отриману інформацію і зберігає її до бази даних в зручному для перегляду іншими елементами інформаційної системи вигляді. Сервер забезпечує безперервний доступ до даних для мобільного пристрою. І третій елемент системи це мобільний пристрій. Він дає змогу переглядати і взаємодіяти з елементами будинку в зручній для цього формі. Симбіоз цих елементів дає змогу поліпшили умови проживання в оселі.

На підставі проведених розрахунків інформаційна система була розроблена з використанням таких технологій:

- MySQL;
- SQL;
- PHP;
- Java;
- Volley;

– C++;
– AVR libc;
– SPI.h
– Ethernet.h;

MySQL — вільна система керування реляційними базами даних, яка була розроблена для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом.

SQL — це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також керування базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення і вилучення даних за допомогою використання системи керування і адміністративних функцій. SQL також включає CLI (Call Level Interface) для доступу і керування базами даних дистанційно. Ця архітектура передбачає виділення однієї з машин мережі як головної (сервер). На такій машині зберігається спільна централізована БД. Усі інші машини мережі виконують функції робочих станцій, за допомогою яких підтримується доступ користувацької системи до бази даних. Файли бази даних відповідно до призначених для користувача запитів передаються на робочі станції, де в основному і проводиться обробка даних. Користувачі також можуть створювати на робочих станціях локальні БД, які використовуються ними монопольно.

PHP — мова, код якої можна вбудовувати безпосередньо в html-код сторінок, які, у свою чергу, коректно оброблюватимуться PHP-інтерпретатором. Обробник PHP просто починає виконувати код після відкриваючого тегу (`<?php`) і продовжує виконання до того моменту, поки не зустріне закриваючий тег. Велика різноманітність функцій PHP дає можливість уникати написання багаторядкових функцій, призначених для користувача, як це відбувається в C або Pascal.

Java — мова програмування загального призначення. Відноситься до об'єктно-орієнтованих мов програмування, до мов із сильною типізацією. Додаток який виступає в ролі дистанційного пульта керування має бути мобільним. Мова програмування Java чудово підходить для написання додатків для мобільних пристроїв на базі ОС Android.

Volley — це бібліотека, яка робить мережеві додатки для Android простішими і найголовніше швидшими. Вона керує обробкою і кешуванням мережевих запитів, і це економить дорогоцінний час розробників від опису одного і того ж коду мережевого запиту/розрахунку з кешу знову і знову. І ще одна перевага, менше коду, менша кількість помилок.

Переваги використання Volley:

- Volley автоматично складає всі мережеві запити.
- Volley буде приймати на себе всі запити ваших додатків виконувати їх для вилучення відповідей або зображень із веб-сайтів.
- Volley забезпечує прозорість дискового кешування і кешування в пам'яті.
- Volley забезпечує потужне API для заміни запиту. Можна скасувати один запит або встановити кілька запитів для відмін.
- Volley забезпечує потужні можливості зміни.
- Volley надає інструменти відладки та трасування.

C++ — компілований, статично типізована мова програмування загального призначення. isocpp.org (англ.) Підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування. Мова програмування пристроїв Arduino заснована на C/C++ і скомпонована з бібліотекою AVR Libc і дозволяє використовувати будь-які її функції. Разом з тим він простий у освоєнні, і зараз Arduino — це, мабуть, найзручніший спосіб програмування пристроїв на мікроконтролерах. Мове Arduino можна розділити на чотири розділи оператори, дані (змінні та константи), функції та бібліотеки. Пакет AVR Libc надає підмножину стандартної бібліотеки C для 8-розрядних мікроконтролерів RISC Atmel AVR. Крім того, бібліотека надає базовий код запуску, необхідний більшості програм.

Послідовний периферійний інтерфейс (SPI) — це послідовний синхронний протокол передачі даних, використовуваних мікроконтролерами для обміну даними з одними або декількома периферійними пристроями на невеликих відстанях.

Ethernet.h — дана бібліотека дозволяє Arduino виходити в Інтернет за допомогою плат розширення Arduino Ethernet. При цьому Arduino може виступати як у ролі сервера, що приймає вхідні з'єднання, так і клієнтом, що з'єднується з видаленим сервером. Бібліотека підтримує до 4 одночасних з'єднань (вхідних, вихідних, або і тих, і інших).

2.4.Опис структури системи та алгоритмів її функціонування.

Щоб розібрати систему по складовим треба виявнити з яких елементів складається інформаційна система. Розроблена інформаційна система складеться з:

– контролер на базі мікропроцесора Arduino;

– сервер;

– мобільний додаток на базі ОС Android 7.0 або вище;

Розглянувши спрощену схему наведену в Додатку А і в Додатку Б (рис. 2.3, рис. 2.4) розберемо інформаційну систему по компонентам.

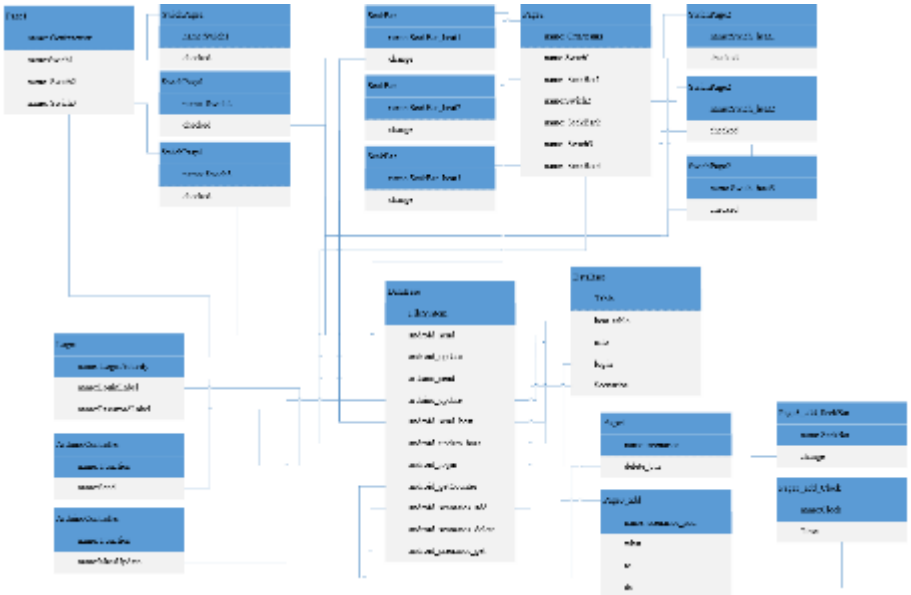


Рис 2.3 UML діаграма класів інформаційної системи

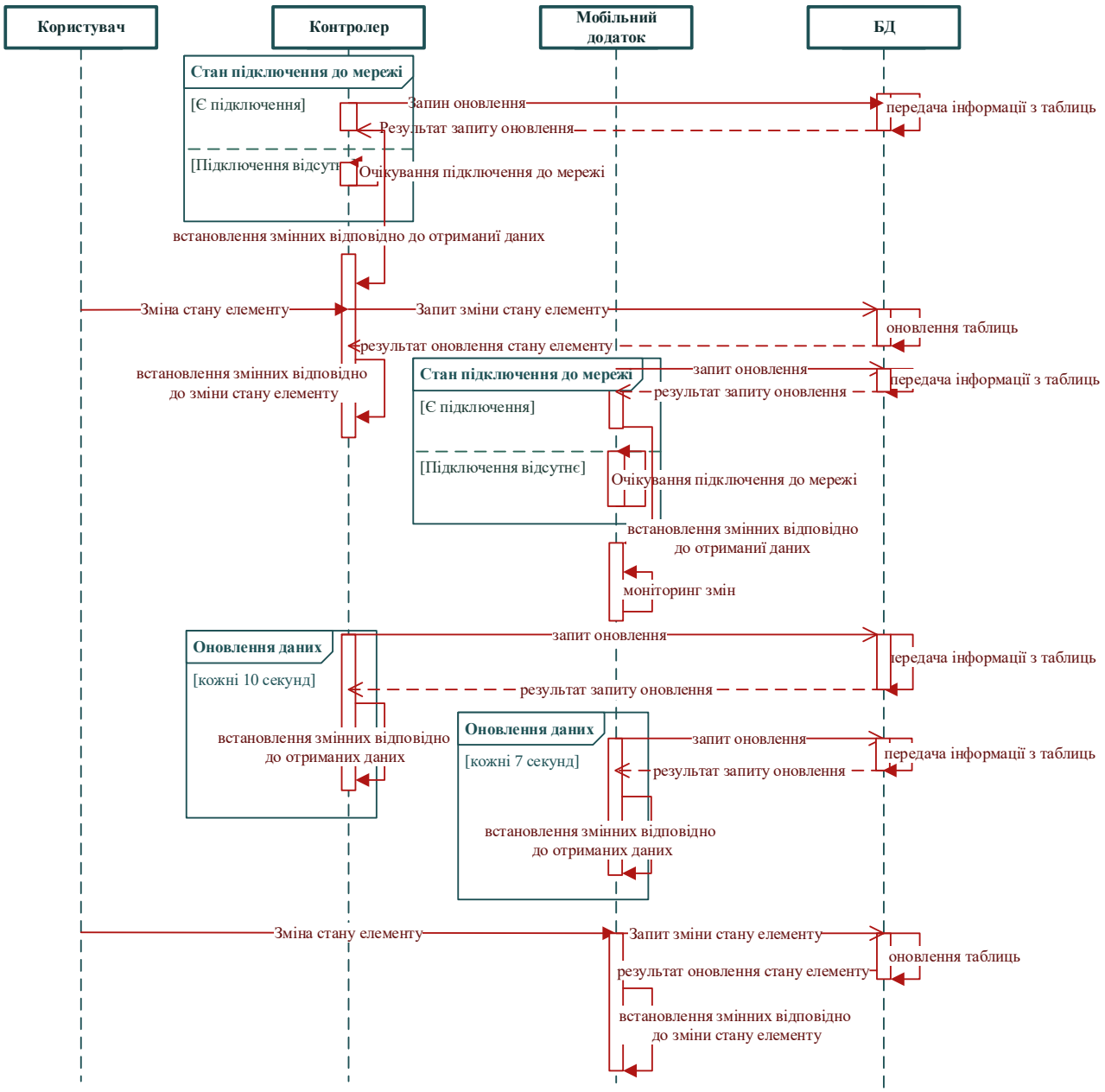


Рис 2.4 UML діаграма послідовностей інформаційної системи

Розглянемо перший елемент інформаційної системи, це мікроконтролер. На базі цього мікроконтролеру побудована логічна частина інформаційної системи. Мікроконтролер відповідає за ввімкненням або вимиканням елементів розумного будинку, займається моніторингом температури та зв’язується з сервером для занесення цієї інформації до бази даних. На рисунку 2.5 зображена повна схема розташувань виводів мікросхеми Arduino MEGA 2560 rev3 (pins, піни). Всього виводів у мікросхеми 53шт. Звісно для демонстраційної моделі не були використані всі виводи, але якщо реалізовувати інформаційну систему в

повному обсязі для поетку великого будинку то їх не буде достатньо в повному обсязі, для цього потрібно буде підключити ще один мікроконтролер через Serial Port, під'єднати вихід на вхід (на рисунку 2.5 зображені як RX0 вхід і TX0 вихід) аналогічно і для другого контролера.

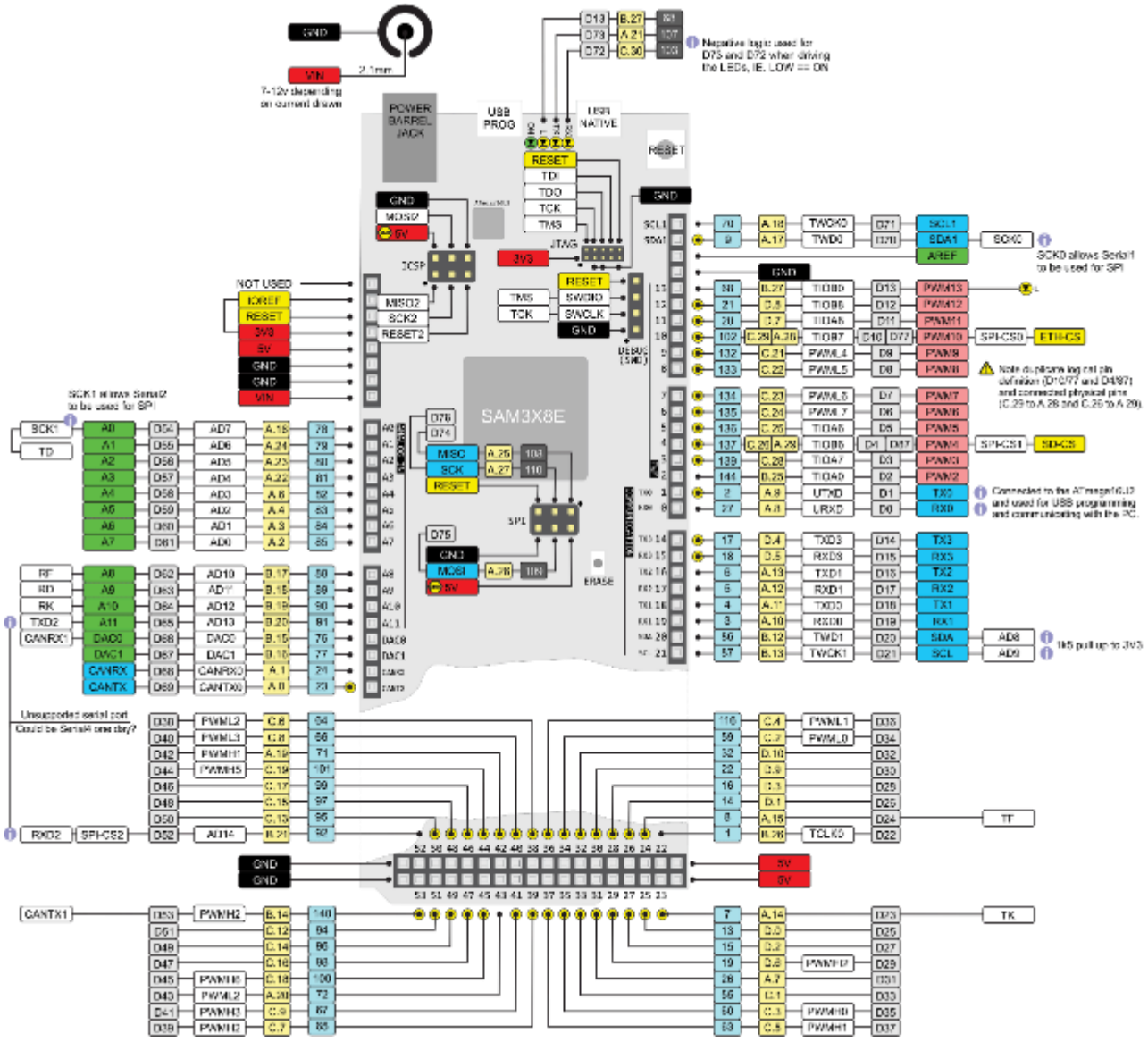


Рис 2.5 Повна схема розташувань виводів, пінів мікросхеми Arduino MEGA 2560 rev3

Алгоритм функціонування програми мікроконтролера дуже простий. По перше при запуску мікроконтролера програма реєструє веб-сервер шилд Ethernet Shield W5100 в локальній мережі по завідомо пустій IP-адресі, наприклад 192.168.1.8, задає шилду MAC-адресу в шістнадцятирічній системі відліку наприклад 0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02. По друге один раз ініціалізуються всі глобальні змінні які приймають участь в керуванні елементами розумного будинку, задаються змінні для таймерів і інтервал

оновлення даних з серверу, створюється структурний елемент (буфер) для зберігання інформації отриманої з серверу. По трете ініціалізується Serial Port для віддзеркалення процесів які йдуть в середині мікроконтролера, ініціалізуються піни при чому треба обов'язково вказати які з них при назначені для входу інформації, а які для виходу на керовані елементи, ініціалізується буфер. По четверте йде запуск веб-сервер шилду Ethernet Shield W5100 в локальній мережі по зареєстрованим адресам IP і MAC, встановлюється затримка в 1 секунду аби шилд встиг ініцалізуватися і в повному обсязі відправив POST-запит на сервер БД. Мікроконтролер звертається до функції в якій прописаний алгоритм відправки запиту на сервер, йде зчитування даних з MySQL бази даних і записується в відповідні змінні в буфері, зчитуються дані з буферу і призначаються для відповідних змінних керованих елементів, після чого призначається отриманий статус керованому елементу відповідно до його змінної в програмі контролеру. Після цих дій контролер запалить вивід під номером 13, він же індикатор готовності до роботи і в Serial Port виведе текстову лінію з написом "READY". Мікроконтролер провів ініціалізацію і готовий до штатного функціонування. При виникненні несправності мікроконтролер не ввімкне вивід під номером 13 і в Serial Port виведе текстову лінію з написом " FALSE update".

Мікроконтролер готовий до роботи, кожні 10 секунд він буде звіряти данні на сервері з станом кожного функціонального елементу. При натисканні на кнопку керування елементу освітлення в його відповідності до підключеного піна управління запускається функція IF яка звіряє змінну FLAG і стан кнопки, якщо кнопка була натиснута і минулого разу стан елемента освітлення був вимкнутий, то змінна FLAG набуває стану вимкнутої, а стан елементу освітлення набуває ввімкнемо. І навпаки, якщо кнопка була натиснута і минулого разу стан елемента освітлення був ввімкнуто, то змінна FLAG набуває стану ввімкнуто, а стан елементу освітлення набуває вимкнемо. Це зроблено для того аби відійти від стандартів перемикачей старого типу, ті які стоять у кожного вдома. Все що

тепер треба це натиснути звичайну кнопку, як на клавіатурі, для ввімкнення так і для вимкнення освітлення.

При зміні стану перемикача кожного разу за допомогою спеціальної функції (sendpin) програми данні про номер піну його стан і стан змінної FLAG відправляються на сервер і зберігаються до бази даних MySQL.

Функція sendpin слугує для відправлення станіу елементу розумного будинку на сервер. В якості аргументів приймає: номер піна, його стан, минулий стан до натискання на кнопку. Після прийому даних функція клієнтське підключення до серверу, виводить в командний рядок напис «SendpinTRUE», строює HTML запит за допомогою методу POST, в якому по черзі заповнює дані до запиту в такому порядку: номер піна, його стан, минулий стан до натискання на кнопку. Функція sendpin закриває підключення до серверу чистить буфер контролеру. Функція має тип void це означає, що функція не повертає ніяких даних. В разі відсутності з'єднання з сервером виводить в командний рядок напис «SendpinFALSE».

Функція updatemass слугує для повного оновлення станів елементів розумного будинку. В якості аргументів приймає: номери пінів, які треба оновити. Після прийому даних ініціалізує структуру Mass massled, створює підключення до мережі інтернет, виводить в командний рядок напис «Sendmass TRUE» і строює HTML запит за допомогою методу POST про ті елементи стан яких треба оновити. Після прийому даних виводить в командний рядок напис «Update...», переводить отримані дані в масив символів, обирає символ, з відповіді, по його порядковому номеру із масиву отриманих в якості відповіді серверу символів і записує його до змінної структури Mass відповідно до отриманих пінів в якості аргументів. Виводить в командний рядок напис з переліком отриманих і записаних даних. Функція updatemass закриває підключення до серверу чистить буфер контролеру. В разі відсутності з'єднання з сервером виводить в командний рядок напис «FALSE update». По завершенню роботи повертає структуру Mass. Як тільки структура буде повернута, контролер призначте зчитані з неї дані для відповідних змінних керованих елементів, після

чого призначається отриманий статус керованому елементу відповідно до його змінної в програмі контролеру.

Функція sendmass слугує для відправлення всіх станів елементів розумного будинку на сервер. В якості аргументів приймає: номер піна, його стан, минулий стан до натискання на кнопку і так далі аналогічно в залежності від кількості пінів. Після прийому даних функція клієнтське підключення до серверу, виводить в командний рядок напис «Sendmass TRUE», строює HTML запит за допомогою методу POST, в якому по черзі заповнює дані до запиту в такому порядку: номер піна, його стан, минулий стан до натискання на кнопку і так далі аналогічно в залежності від кількості пінів. Функція sendmass закриває підключення до серверу чистить буфер контролеру.

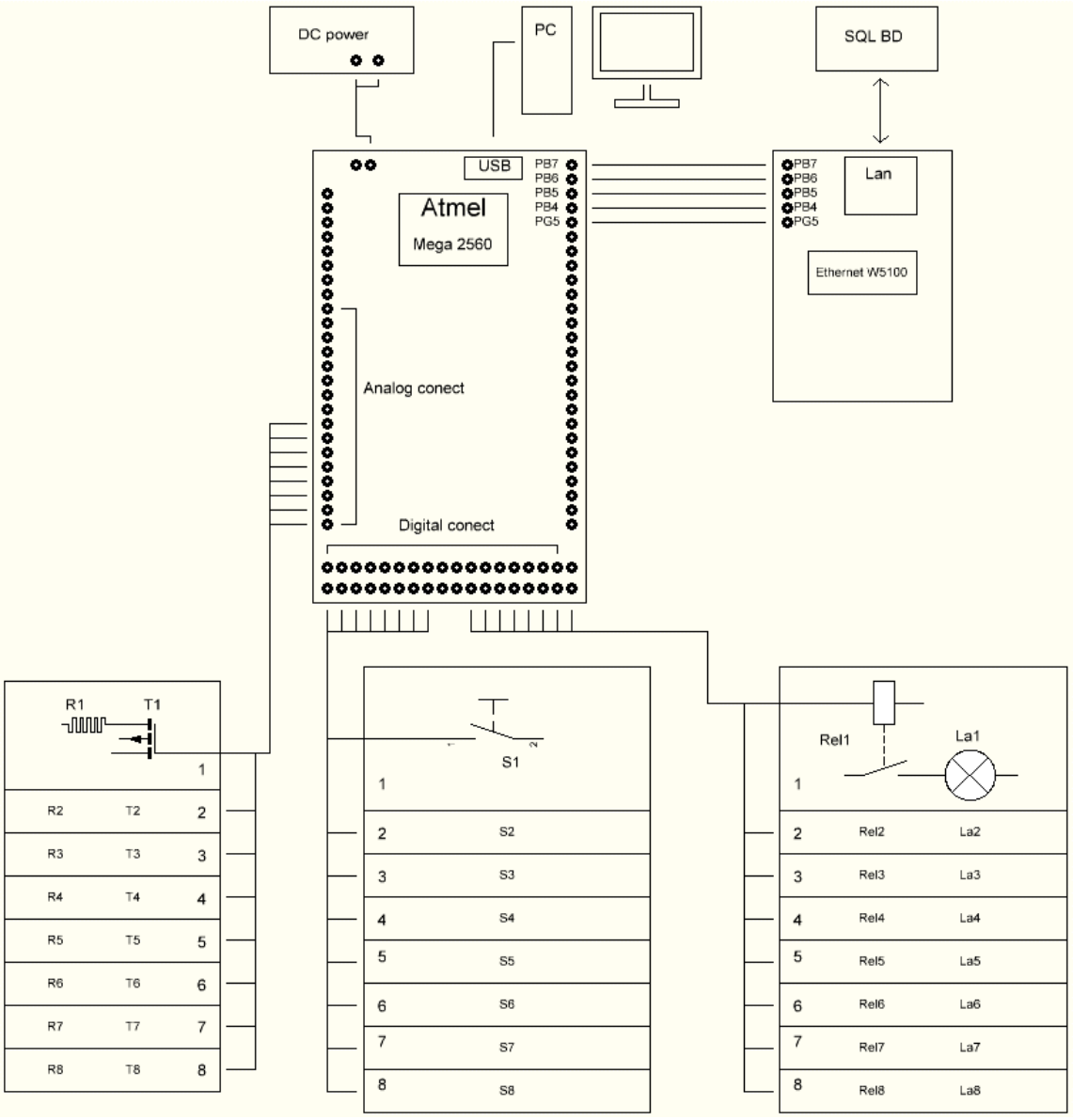


Рис 2.6 Функціональна схема підключення до плати Arduino MEGA 2560.

Функція має тип void це означає, що функція не повертає ніяких даних. В разі відсутності з’єднання з сервером виводить в командний рядок напис «Sendmass FALSE».

Розглянувши функціональну схему підключення елементів до плати Arduino MEGA 2560 (Рис 2.6) ми побачимо підключені керовані елементи (знизу з ліва на право: обігріваючі елементи підключені через потужні напівпровідникові керовані регулятори температури, елементи керування розетками підключені через реле і елементи освітлення підключені через реле), також до мікросхеми підведений постійний струм з напругою 9V/2A, є можливість підключення до персонального комп’ютеру, для відладки і усунення проблем через порт USB type A, і підключений мережевий адаптер веб-сервер шилд Ethernet Shield W5100, який має вихід до мережі інтернет.

Розглянемо другий елемент інформаційної системи(Рис 2.7). На базі безкоштовного хостінгового сервісу побудована серверна частина інформаційної системи. Сервер відповідає за комунікацію користувачів інформаційної системи, її елементів, та збереження одиниць інформації в базі даних. Серверна частина розділяється на:

- базу даних;
- PHP сторінки;

База даних зберігає в розділених таблицях інформацію про стан інформаційної системи. PHP сторінки дають змогу як контролеру так і користувачеві отримувати інформацію про стан інформаційної системи та вносити зміни в стан елементів розумного будинку. В базі даних знаходяться таблиці, а в файловій системі PHP скрипти.

Розглянемо кожен функціональний елемент серверу.

Перший елемент системи який я розгляну це база даних. База даних складеться з таких таблиць:

- login;
- scenarios;
- scenarios_delete;

– heat_hable;

– info;

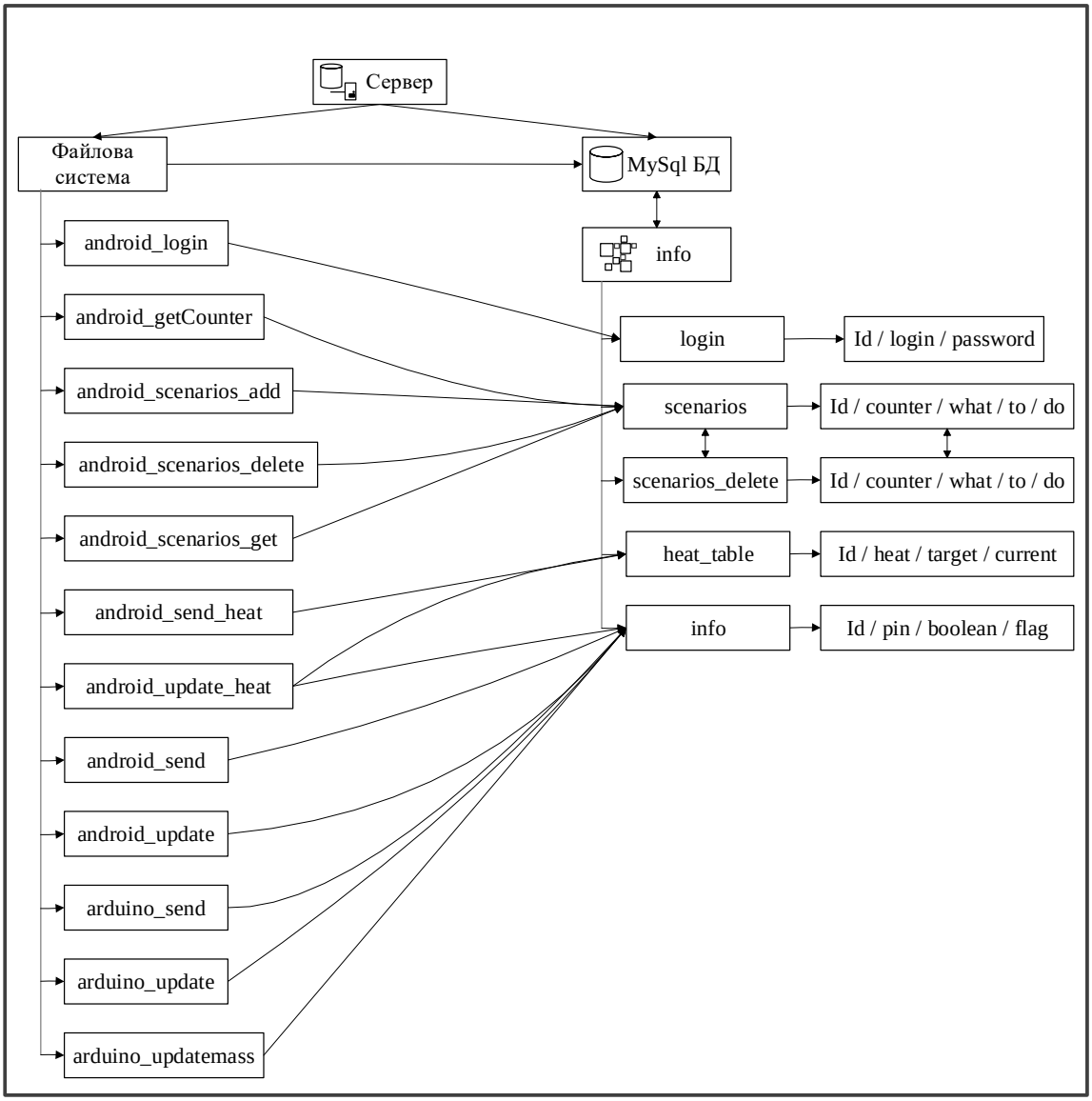


Рис 2.7 Схема архітектури серверної частини інформаційної системи «Розумний дім»

Другий елемент системи — це файлова система, до неї належать всі файли розташовані на сервері, і слугують для взаємодії клієнтів ІС з базою даних. До елементів файлової системи можна віднести:

– android_login.php;

– android_getCounter.php;

– android_scenarios_add.php;

– android_scenarios_delete.php;

– android_scenarios_get.php;

– android_send_heat.php;
– android_update_heat.php;
– android_send.php;
– android_update.php;
– arduino_send.php;
– arduino_update.php;
– arduino_updatemass.php;

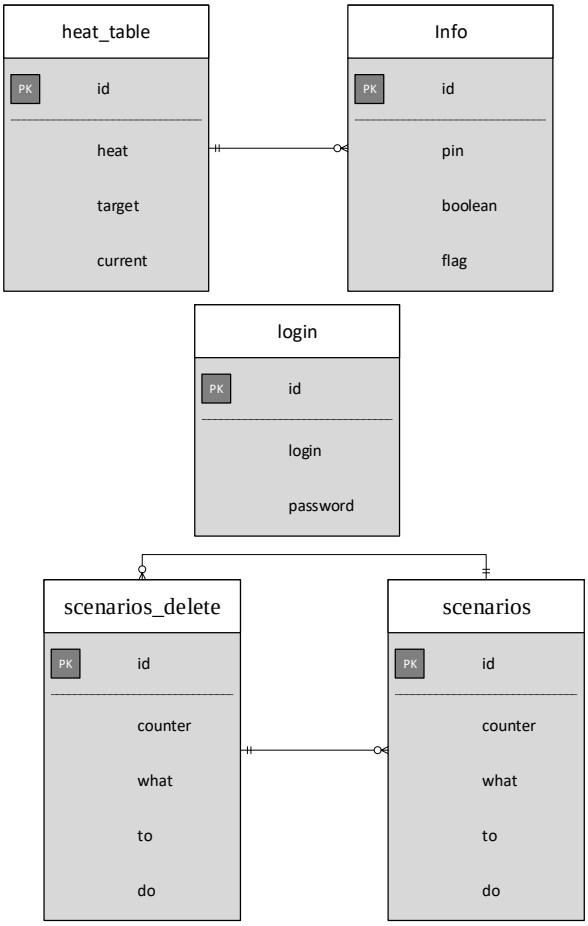


Рис 2.8 ER діаграма БД інформаційної системи «Розумний дім»

Стовпці які належать таблиці login: id, login, password. В рядки цієї таблиці вписані: номер рядка (id), ім'я для входу до мобільного додатку (login), пароль для входу до мобільного додатку (password). Ця таблиця відповідає за зберігання інформації для входу до додатку. За доступ до таблиці відповідає PHP скрипт, який знаходиться в файлі android_login. В якості даних скрипт від мобільного

додатку приймає логін і пароль введені в спеціальні текстові поля, за допомогою SQL команди (SELECT * FROM login WHERE login = '$login') дістає з бази даних всю інформацію про рядок, звіряє текстове поле з БД із отриманим з мобільного додатку, якщо паролі збігаються результатом роботи є код 1. Якщо введеного логіну або паролю не існує скрипт виведе 0.

Таблиця scenarios. До цієї таблиці належать такі стовпці: id, counter, what, to, do. В рядки цієї таблиці вписані: номер рядка збереженого в базі даних (id), порядковий номер сценарію для правильного відображення в мобільному додатку(counter), порядковий номер елементу розумного будинку(what), інформація про те що треба зробити(to), інформація коли або скільки потрібно зробити(do). За доступ до таблиці відповідає декілька PHP скриптів, які знаходиться в файлах: android_getCounter, android_scenarios_add, android_scenarios_delete, android_scenarios_get.

Дані скрипт android_getCounter від мобільного додатку не приймає, при звернені до скрипта він відповідає кількістю елементів які знаходяться в таблиці scenarios і виводить максимальний елемент з стовпця counter, це потрібно для коректного відображення елементів сценарію в мобільному додатку.

Скрипт android_scenarios_add відповідає за додавання нового сценарію до таблиці scenarios. В якості змінних від мобільного додатку він приймає: counter, what, to, do, в кодованому вигляді і створює новий рядок в таблиці scenarios за допомогою SQL команди (INSERT INTO scenarios (counter , What , Too , Doo) VALUES ('$counter', '$What', '$To', '$Do') скрипт нічого не відповідає після виконання.

Скрипт android_scenarios_delete відповідає за видалення сценарію з таблиці scenarios. В якості змінних від мобільного додатку він приймає counter, далі за допомогою SQL команд ("DELETE FROM `scenarios` WHERE `counter` = '$counter'; "), ("UPDATE `scenarios` SET `counter` = `counter` -1 WHERE `counter` > '$counter';") видаляє рядок в якому знаходиться цей сценарій під номером counter, а всі рядки в яких counter більше ніж у видаленого рядку зменшує counter на одиницю.

Скрипт android_scenarios_get відповідає за отримання всіх сценаріїв на Android девайс. Не приймає ніяких даних від додатку. При зверненні до сторінки скрипт за допомогою SQL команди (SELECT * FROM scenarios) з таблиці scenarios дістає всі рядки і виводить їх в сортованому вигляді, а саме додає знаки розділення мінус (-) між елементами стовпців і знак нового рядку (\n) між елементами рядків. Це роблено для коректного прийому даних на стороні мобільного додатку і для зручного сортування даних до масиву, і виводу сценаріїв на екран користувача.

Таблиця scenarios_delete слугує для резервного копіювання сценаріїв в разі виникнення помилки, копіює таблицу scenarios.

Таблиця heat_table слугує для збереження даних температури нагрівальних приладів. До цієї таблиці належать такі стовпці: id, heat, target, current. Стовпець id відповідає за порядковий номер елементу в базі даних, heat відповідає за порядковий номер нагрівального приладу, в target збережена інформація про температуру до якої треба обігрівати кімнату і в стовпці current збережена інформація про поточну температуру. За доступ до таблиці відповідає декілька PHP скриптів, які знаходиться в файлах: android_send_heat, android_update_heat.

PHP скрипт android_send_heat відповідає за оновлення інформації про цільову температуру в таблиці heat_table, в якості змінної приймає номер нагрівального приладу і цільову температуру. За допомогою SQL команди (UPDATE heat_table SET target = '$target' WHERE heat = '$heat') оновлює дані в таблиці і вписує до неї отримані дані. По завершенню роботи скрипт не відповідає ніякими даними.

Скрипт android_update_heat слугує для оновлення інформації про обігрівальні прилади в мобільному додатку. В якості змінних приймає номери нагрівальних приладів, для яких потрібне оновлення. За допомогою SQL команди (SELECT * FROM heat_table WHERE heat = '$heat'), скрипт обирає з всього рядку поточну температуру в кімнаті, а за допомогою SQL команди (SELECT * FROM info WHERE pin = '$pin') з таблиці info обирає стан приладу (увімкнено або вимкнено) і передає клієнту. Це зроблено для одночасного

оновлення як стану нагрівального приладу так і для оновлення поточної температури.

Таблиця info слугує для збереження даних про всі електроприлади в інформаційній системі, а саме про їх стан (увімкнено або вимкнено). До цієї таблиці належать такі стовпці: id, pin, boolean, flag. В рядки таблиці записані данні про: порядковий номер(id), номер електроприладу в контролері (пін, pin), стан приладу (boolean), і його минулий стан (flag). За доступ до таблиці відповідає декілька PHP скриптів, які знаходиться в файлах: android_send, android_update_heat, arduino_send, arduino_update, arduino_updatemass.

Скрипт android_update_heat слугує для оновлення інформації про поточну і цільову температуру. Ніяких даних від додатку не приймає, за допомогою SQL команди (SELECT * FROM info WHERE pin = '$pin') з таблиці info обирає стан приладу (увімкнено або вимкнено) і передає клієнту.

Скрипт android_send слугує для оновлення даних в таблиці info від Android додатку. Приймає від мобільного додатку інформацію: номер приладу, його стан, і його минулий стан, і за допомогою SQL команди (UPDATE info SET boolean = '$bool' , flag = '$flag' WHERE pin = '$pin'), оновлює дані в потрібному рядку таблиці info. По завершенню роботи скрипт не передає ніякої інформації.

Скрипт android_update слугує для отримання інформації мобільним додатком. Приймає від мобільного додатку інформацію дані номеру приладу про який потрібно оновити інформацію. За допомогою SQL команди (SELECT * FROM info WHERE pin = '$pin') обирає з таблиці info інформацію про стан конкретного приладу і виводить її для відображенням мобільним додатком.

Скрипт arduino_send слугує для оновлення даних таблиці info від мікроконтролера Arduino. Приймає від мікроконтролера таку інформацію: номер приладу, його стан, і його минулий стан, і за допомогою SQL команди (UPDATE info SET boolean = '$bool' , flag = '$flag' WHERE pin = '$pin'), оновлює дані в потрібному рядку таблиці info. По завершенню роботи скрипт не передає ніякої інформації.

Скрипт arduino_update слугує для отримання інформації мікроконтролером. Приймає від мікроконтролера інформацію, дані номеру приладу про який потрібно оновити інформацію. За допомогою SQL команди (SELECT * FROM info WHERE pin = '$pin') обирає з таблиці info інформацію про стан конкретного приладу і виводить її для переключення реле елементів розумного будинку.

Скрипт arduino_updatemass слугує для отримання інформації мікроконтролером але на відміну від скрипту arduino_update оновлює всі елементи, в іншому аналогічний скрипту arduino_update.

Розглянемо третій елемент інформаційної системи — це Android додаток. Додаток розроблено на мові програмування Java в середовищі розробки Android studio, з використанням бібліотеки Volley. Бібліотека слугує для обміну даними між сервером і клієнтом. На рисунку 2.9 зображено архітектуру мобільного додатку.

При відкритті мобільного додатку на екрані мобільного пристрою з'являється Activity_login. В даній активності є два поля, одне для вводу текстової інформації (login), друге для вводу цифр (password), кнопка для здійснення авторизації до мобільного додатку і CheckBox. CheckBox при встановленню прапора в цьому полі дані з полів логіну і паролю зберігаються до файлової системи в папку самого додатку в txt файлі. При натисканні на кнопку авторизації додаток відправляє дані на сервер, де спеціальний android_login.php скрипт оброблює їх, і в разі правильного вводу логіну і паролю переводить на головний екран MainActivity. На екрані MainActivity знаходиться контейнер з фрагментами екранів і контейнер tab_layout з переліком сторінок вгорі екрану, для навігації треба натиснути на елемент tab_layout або використовувати «свайп». Для визначення того який фрагмент потрібно відображати на екрані, написаний Java код в файлі ViewPagerAdapter.java. В якості змінної код приймає позицію обраного положення tab_layout і повертає fragment який потім вставляється в контейнер view_pager для відображенню на передньому плані екрану додатку. Всього таких fragment є три це: fragment_page1, fragment_page2, fragment_page3.

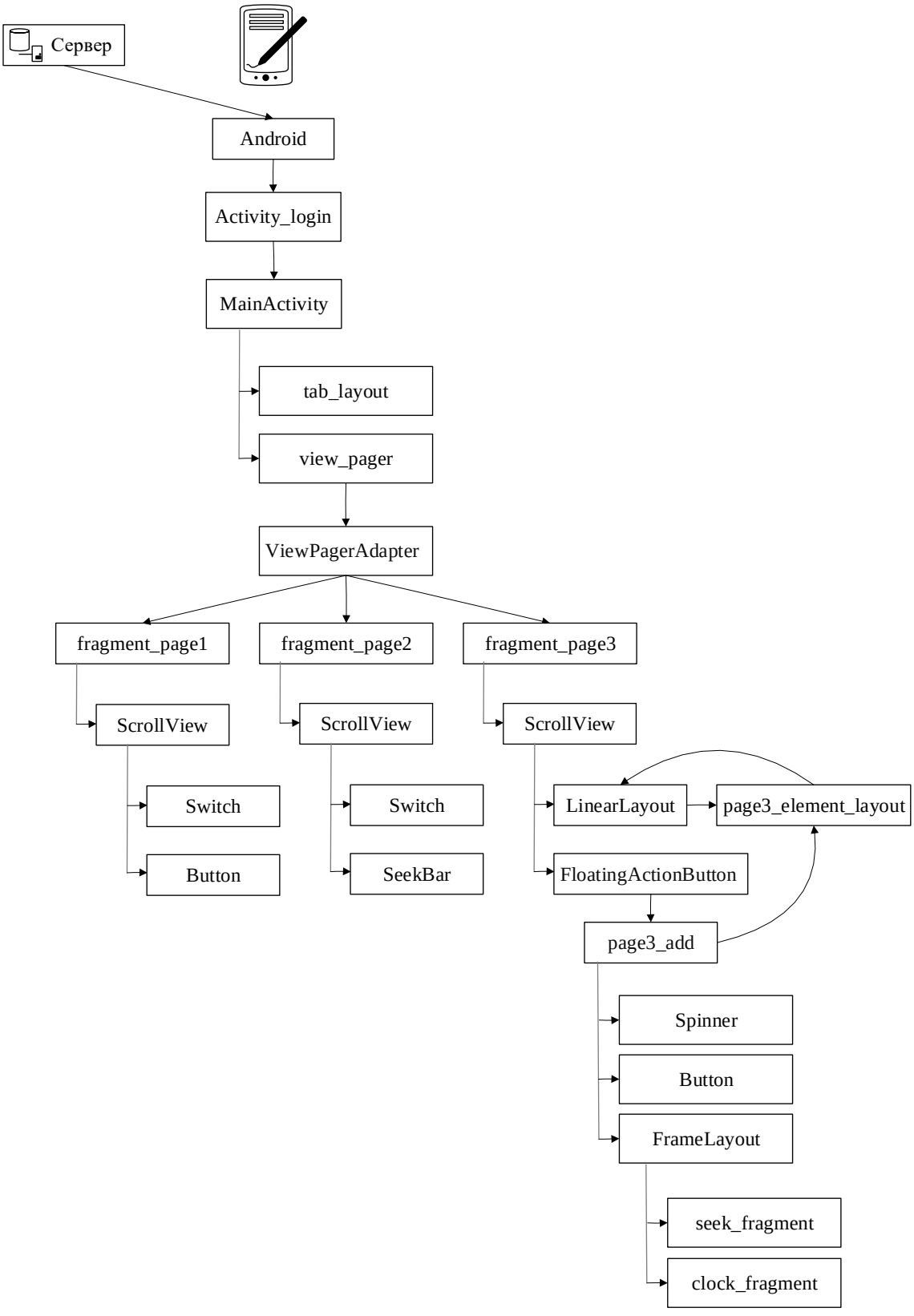


Рис 2.9 Схема елементів мобільного додатку «Розумний дім»

Треба зауважити, що при відкритті фрагменту за допомогою бібліотеки Volley додаток оновлює інформацію про елементи які знаходяться на екрані. В

кожного фрагменту цей код унікальний так як кожен з них відображає різні елементи управління елементами інформаційної системи «Розумний будинок».

fragment_page1 слугує для відображення інформації про освітлювальні прилади, на екрані fragment знаходяться такі елементи ScrollView, Switch, Button. ScrollView використовується для плавного перегляду елементів які знаходяться за межами екрану за допомогою свайпу вгору або вниз. В середині цього контейнеру знаходяться елементи Switch і Button. Елементів Switch в цьому контейнері знаходиться стільки скільки освітлювальних приладів підключено до інформаційної системи. Елемент Switch використовується користувачем як для перегляду інформації про стан приладу, програмно зв’язаного з ним так і для зміни його стану. При натисканні на Switch в додатку змінюється його стан, відповідно протилежно до того який в нього був, при цьому данні про змінення стану за допомогою бібліотеки Volley відправляються на сервер на сторінку android_send.php де записуються до бази даних. При натисканні Button, яка відповідає за зміну стану всіх елементів на протилежні значення, всі елементи Switch які знаходяться на екрані fragment_page1 змінюють свій стан на протилежний, інформація про зміну стану відправляється на сервер за допомогою бібліотеки Volley на сторінку android_send.php де записуються до бази даних. Для оновлення інформації про елементи інформаційної системи використовує скрипт android_update.php. При відсутності інтернет з’єднання додаток виведе тост про помилку оновлення або відправлення даних.

fragment_page2 слугує для відображення інформації про прилади обігріву будинку, на екрані fragment знаходяться такі елементи ScrollView, Switch, SeekBar. ScrollView використовується для плавного перегляду елементів які знаходяться за межами екрану за допомогою свайпу вгору або вниз. В середині цього контейнеру знаходяться елементи Switch і SeekBar. Елементів Switch в цьому контейнері знаходиться стільки скільки приладів опалення підключено до інформаційної системи. Елемент Switch використовується користувачем як для перегляду інформації про стан приладу, програмно зв’язаного з ним так і для зміни його стану. При натисканні на Switch в додатку змінюється його стан,

відповідно протилежно до того який в нього був, при цьому данні про змінення стану за допомогою бібліотеки Volley відправляються на сервер на сторінку android_send.php де записуються до бази даних в відповідну таблицю. Також в текстовому полі елементу Switch знаходиться інформація про поточну температуру в будинку. SeekBar слугує для змінення і встановлення цільової температури повітря в кожній кімнаті, відповідно до програмо підключеного до нього елементу. При початку змінення температури зліва від елементу в текстовому полі відображається та температура яка буде встановлена, а при завершенні встановлення температури і відпусканню елементу дані положення повзунка зчитуються та формуються в пакет для відправки на сервер за допомогою бібліотеки Volley на сторінку android_send_heat.php. Також на фрагменті знаходиться SeekBar який змінює стан всіх SeekBar на встанновлений в цьому SeekBar. При оновлені даних про стан перемикачів додаток бере інформацію з android_update.php, а для оновлення поточної і цільової температури з android_update_heat.php. При відсутності інтернет з’єднання додаток виведе тост про помилку оновлення або відправлення даних.

fragment_page3 слугує для відображення інформації про сценарії встановлених всіма користувачами. На екрані fragment знаходяться такі елементи: ScrollView, LinearLayout, FloatingActionButton. При відкритті екрану додаток бере інформацію з android_scenarios_get.php для відображення сценаріїв. Для коректного відображення сценаріїв додатку потрібно знати кількість елементів для відображення, для цього в MainActivity.java описана функція котра за допомогою бібліотеки Volley і скрипту android_getCounter.php отримує кількість елементів записаних в базі даних. Також в MainActivity.java описана функція котра за допомогою бібліотеки Volley і скрипту android_scenarios_get.php отримує всі сценарії, за допомогою циклу FOR розкриває масив даних по рядкам і вкладеного циклу FOR розкриває масив даних по стовпцям після чого отриманий масив записується до внутрішнього масиву і при відкритті fragment_page3 відображається в контейнері LinearLayout, при цьому в контейнері знаходяться динамічно додані елементи які відображають

дійсну картину даних в кожного користувача в реальному часі. Динамічне додавання елементів можливе завдяки створенню загального шаблону відображення отриманих даних. Для реалізації цієї функції створюється сцена, для кожного рядку масива, присвоюється унікальний id(id сцени збігається з порядковим номером в таблиці scenarios), знаходяться всі елементи шаблону і привласнюються індексові імена, кожному елементу шаблону привласнюються отримані дані того рядку внутрішнього масиву з яким збігається лічильник і сформована сцена відображається на екрані користувача. Для видалення сценарію в кожному шаблоні є кнопка для видалення елементу з бази даних, для цього додаток зчитує id натиснутого елементу і за допомогою бібліотеки Volley відправляє дані порядкового номеру до скрипту android_scenarios_delete.php після чого здійснюються видалення і оновлення лічильників як в додатку так і в базі даних. Також на фрагменті fragment_page3 знаходиться FloatingActionButton, вона знаходиться завжди в нижньому правому кутку екрану. При натисканні на кнопку відкривається нове activity page3_add.

page3_add слугує для створення нових сценаріїв по шаблону. При відкритті на екрані користувача з’являються такі функціональні елементи: Spinner, Button, FrameLayout. В першому елементі Spinner відображається елементи стан яких треба змінити. В другому елементі Spinner відображається той стан який треба встановити, при цьому встановити температуру для освітлювального приладу неможливо, для рішення цієї проблеми, елементи першого Spinner поділені на категорії, і для обігріваючих приладів є свій випадаючий список в якому можна обрати температуру. FrameLayout слугує для відображення елементів: seek_fragment і clock_fragment. При цому при виборі освітлювального приладу відображається clock_fragment для встановлення часу ввімкнення або вимкнення, а при виборі обігрівального приладу в першому випадаючому списку Spinner і виборі встановлення температури у другому Spinner в контейнер FrameLayout завантажується seek_fragment. При натисканні на Button (ОК) данні зчитуються і за допомогою бібліотеки Volley відправляються на сервер, activity

page3_add закривається. При натисканні на Button (CANCEL) activity page3_add закривається.

2.5.Обґрунтування та організація вхідних та вихідних даних програми

Так як інформаційна система складається декількох елементів, треба розібрати вхідні і вихідні дані для кожного з них. Для розробленої інформаційної системи існує дві лінії зв’язку:

– апаратна;

– програмна;

Мікроконтролер Arduino приймає дані як по програмній так і апаратній лінії зв’язку. Апаратні дані мікроконтролер приймає від перемикачів і термометрів, вони слугують для змінення стану підключених до нього керованих елементів, мікроконтролер ніяк не може впливати на ці дані, він може тільки зчитувати дії і стани елементів. В якості зв’язку по програмній лінії керування, контролер підключений до бази даних з якою обмінюється інформацією про стан керованих елементів, при цьому контролер може впливати на значення цих даних, звісно тільки в тих випадках коли користувач інформаційної системи буде змінювати стани по апаратній лінії зв’язку. В якості вхідних даних контролер від серверу приймає стани керованих елементів при оновленні інформації, а в якості вихідних передає значення змінних відповідних керованих елементів.

Android додаток обмінюється даними з сервером за допомогою бібліотеки Volley, він має лише програмну лінію зв’язку. Кожен екран програми підключається до власних скриптів на сервері і отримує від них відповідь. Дані для віх екранів додаток приймає у вигляді рядку String, де дані строго типізовані і йдуть по черзі один за одним, після чого конвертується в масив символів і завдяки строгій типізації даних кожен елемент символьного масиву присвоюється елементам екранів. Окрім MainActivity, ця частина додатку, від сервера, приймає одразу цілий масив даних, після чого за допомогою циклу FOR

розкриває масив даних по рядкам і вкладеного циклу FOR розкриває масив даних по стовпцям після отриманий масив записується до внутрішнього масиву.

Сервер в своїй сутності є як базою даних так і посередником в зв’язку між мікроконтролером і мобільним пристроєм. За допомогою PHP скриптів і SQL запитів постачає потрібну інформацію до клієнтів інформаційної системи.

Дані скрипт android_getCounter від мобільного додатку не приймає, при звернені до скрипта він відповідає кількістю елементів які знаходяться в таблиці scenarios і виводить максимальний елемент з стовпця counter.

Скрипт android_scenarios_add відповідає за додавання нового сценарію до таблиці scenarios. В якості змінних від мобільного додатку він приймає: counter, what, to, do, в кодованому вигляді і створює новий рядок в таблиці scenarios за допомогою SQL команди (INSERT INTO scenarios (counter , What , Too , Doo) VALUES ('$counter', '$What', '$To', '$Do') скрипт нічого не відповідає після виконання.

Скрипт android_scenarios_delete відповідає за видалення сценарію з таблиці scenarios. В якості змінних від мобільного додатку він приймає counter, далі за допомогою SQL команд ("DELETE FROM `scenarios` WHERE `counter` = '$counter'; "), ("UPDATE `scenarios` SET `counter` = `counter` -1 WHERE `counter` > '$counter';") видаляє рядок в якому знаходиться цей сценарій під номером counter, а всі рядки в яких counter більше ніж у видаленого рядку зменшує counter на одиницю.

Скрипт android_scenarios_get відповідає за отримання всіх сценаріїв на Android девайс. Не приймає ніяких даних від додатку. При зверненні до сторінки скрипт за допомогою SQL команди (SELECT * FROM scenarios) з таблиці scenarios дістає всі рядки і виводить їх в сортованому вигляді, а саме додає знаки розділення мінус (-) між елементами стовпців і знак нового рядку (\n) між елементами рядків.

PHP скрипт android_send_heat відповідає за оновлення інформації про цільову температуру в таблиці heat_table, в якості змінної приймає номер нагрівального приладу і цільову температуру. За допомогою SQL команди

(UPDATE heat_table SET target = '$target' WHERE heat = '$heat') оновлює дані в таблиці і вписує до неї отримані дані. По завершенню роботи скрипт не відповідає ніякими даними.

Скрипт android_update_heat слугує для оновлення інформації про обігрівальні прилади в мобільному додатку. В якості змінних приймає номери нагрівальних приладів, для яких потрібне оновлення. За допомогою SQL команди (SELECT * FROM heat_table WHERE heat = '$heat'), скрипт обирає з всього рядку поточну температуру в кімнаті, а за допомогою SQL команди (SELECT * FROM info WHERE pin = '$pin') з таблиці info обирає стан приладу (увімкнено або вимкнено) і передає клієнту.

Скрипт android_update_heat слугує для оновлення інформації про поточну і цільову температуру. Ніяких даних від додатку не приймає за допомогою SQL команди (SELECT * FROM info WHERE pin = '$pin') з таблиці info обирає стан приладу (увімкнено або вимкнено) і передає клієнту.

Скрипт android_send слугує для оновлення даних в таблиці info від Android додатку. Приймає від мобільного додатку інформацію: номер приладу, його стан, і його минулий стан, і за допомогою SQL команди (UPDATE info SET boolean = '$bool' , flag = '$flag' WHERE pin = '$pin'), оновлює дані в потрібному рядку таблиці info. По завершенню роботи скрипт не передає ніякої інформації.

Скрипт android_update слугує для отримання інформації мобільним додатком. Приймає від мобільного додатку інформацію дані номеру приладу про який потрібно оновити інформацію. За допомогою SQL команди (SELECT * FROM info WHERE pin = '$pin') обирає з таблиці info інформацію про стан конкретного приладу і виводить її для відображенням мобільним додатком.

Скрипт arduino_send слугує для оновлення даних таблиці info від мікроконтролера Arduino. Приймає від мікроконтролера таку інформацію: номер приладу, його стан, і його минулий стан, і за допомогою SQL команди (UPDATE info SET boolean = '$bool' , flag = '$flag' WHERE pin = '$pin'), оновлює дані в потрібному рядку таблиці info. По завершенню роботи скрипт не передає ніякої інформації.

Скрипт arduino_update слугує для отримання інформації мікроконтролером. Приймає від мікроконтролера інформацію, дані номеру приладу про який потрібно оновити інформацію. За допомогою SQL команди (SELECT * FROM info WHERE pin = '$pin') обирає з таблиці info інформацію про стан конкретного приладу і виводить її для переключення реле елементів розумного будинку.

2.6.Опис розробленої системи

2.6.1 Використані технічні засоби.

Для роботи мобільного додатку необхідна платформа на базі операційної системи ОС Android 7.0 або вище з доступом в інтернет.

Програма для контролеру розроблена спеціально під мікроконтролер на базі Arduino MEGA 2560 rev3 з використанням веб-сервер шилд Ethernet Shield W5100 Arduino. Для коректної роботи серверу потрібно:

- від 150 гігабайт SSD;
- ЦП: Pentium 4;
- 512 мегабайт відеопам’яті;
- 4 гігабайти оперативної пам’яті;
- доступ в інтернет на швидкості від 10 MB/s;

При розробці та тестуванні системи була використана клієнтська персональна ЕОМ з наступними характеристиками:

- Процесор AMD FX-6350 кількість ядер: 6, 3,9 ГГц;
- 12 гігабайт ОЗУ;
- 2 гігабайти відеопам’яті;
- 1 терабайт ППЗ;

Тестування мобільного додатку проводилося на таких операційних системах:

- ОС Android 7.0 на базі оболонки MIUI 11;
- ОС Android 10 на базі оболонки MIUI 12;
- ОС Android 11 на базі оболонки MIUI 12.5;

– ОС Android 12 на базі оболонки MIUI 13;

2.6.2 Використані програмні засоби.

Для створення серверу було обрано безкоштовний хостінговий сервіс www.000webhost.com, даний сервіс забезпечив створення бази даних за допомогою сервісу PhpMyAdmin і віддалене сховище PHP скриптів.

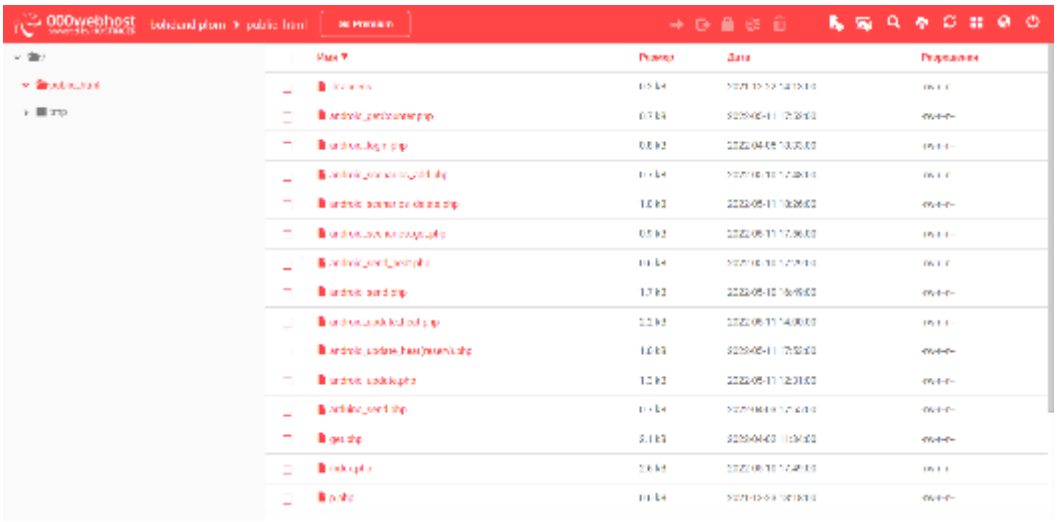


Рис 2.10 Інтерфейс файлової системи www.000webhost.com

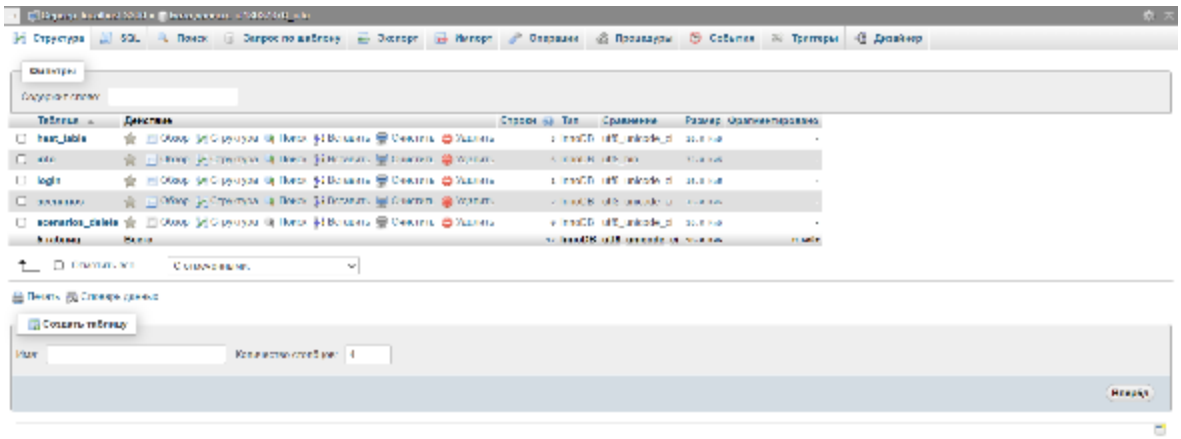


Рис 2.11 Інтерфейс PhpMyAdmin

Для розробки мобільного додатку я використав Android Studio Bumblebee (2021.1.1 patch 2) – в середовищі розробки на основі IntelliJ IDEA, що надає

інтегровані інструменти для розробки та налагодження додатків для платформі Android.

Для розробки програми для мікропроцесора Arduino MEGA 2560 я використав спеціально розроблене середовище програмування Arduino IDE (1.18.1).

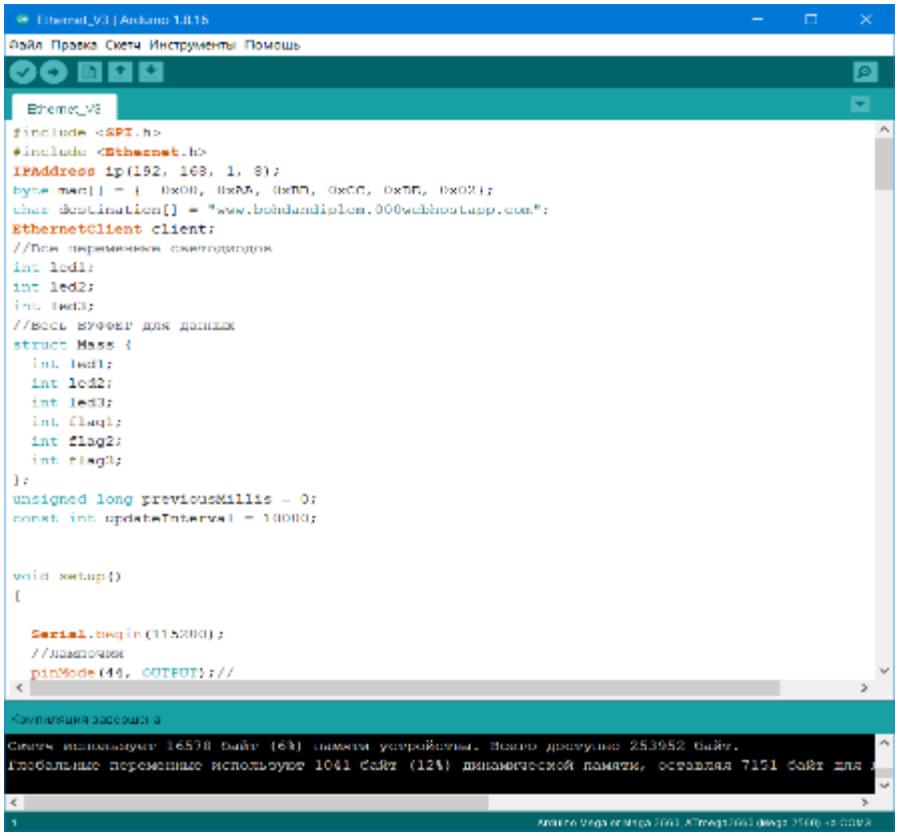


Рис 2.12 інтерфейс Arduino IDE

2.6.3 Виклик та завантаження програми.

Для забезпечення роботи інформаційної системи, по перше треба під’єднати всі піни описані в програмі до відповідних роз’ємів плати мікропроцесора Arduino MEGA 2560, під’єднати шилд Ethernet Shield W5100 Arduino в порт Ethernet під’єднати інтернет шнур підключений до маршрутизатора і подати живлення на плату. Загрузити мобільний додаток та встановити його на пристрій. Після чого інформаційна система буде повністю скомплектована і готова до роботи.

2.6.4. Опис інтерфейсу користувача.

Сервер не має інтерактивного інтерфейсу для користувача, але має HTML сторінку в якій в сортованому вигляді виведена інформація з бази даних.

Получено объектов: 6

ID	PIN	BOOL	FLAG
3	pin1	0	0
4	pin2	0	0
5	pin3	0	0
6	pin4	1	0
1	pin5	1	0
2	pin6	0	0

Получено объектов: 3

ID	HEAT	TARGET	CURRENT
1	1	43	20
2	2	21	20
3	3	45	20

Получено объектов: 2

ID	COUNTER	WHAT	TO	DO
39	0	0	0	0:30
40	1	0	0	7:10

Рис 2.13 HTML сторінка в якій виведена інформація з бази даних

Мікроконтролер Arduino має фізичний інтерфейс користувача, можете поглянути на нього в власному будинку, він нічим не буде відрізняться.

Інтерфейс мобільного додатку має наступний вигляд, для демонстрації адаптивності інтерфейсу скріншоти будуть представлені з телефонів з різною діагоналлю екрану (Рис 2.14 – 2.35).

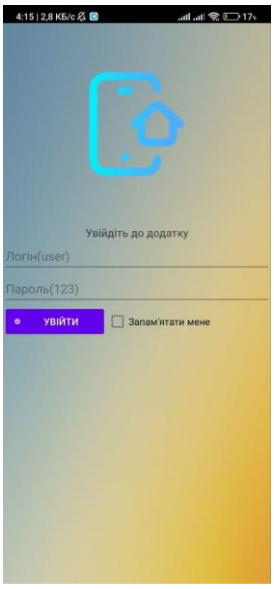


Рис 2.14 Інтерфейс користувача, сторінка входу

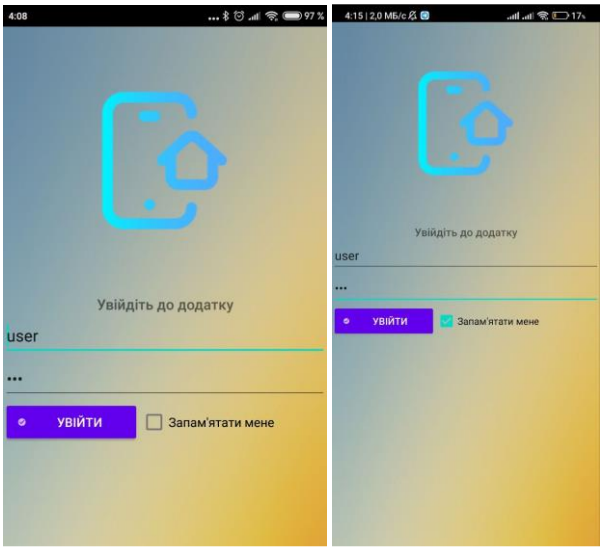


Рис 2.15 Інтерфейс користувача, сторінка входу з заповненими даними

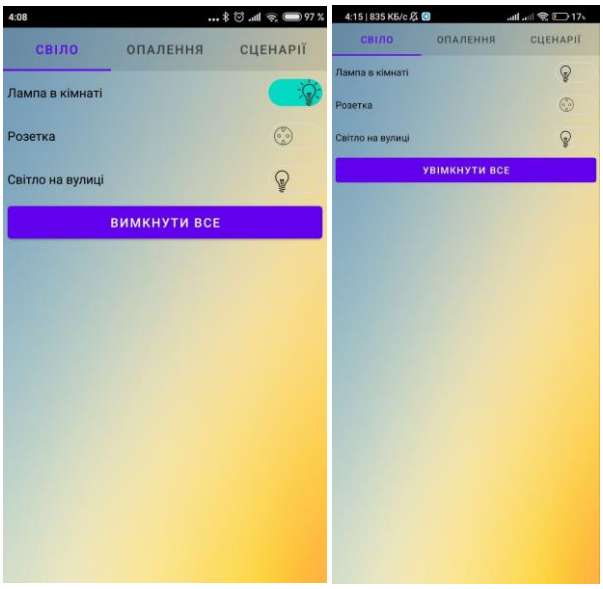


Рис 2.16 Інтерфейс користувача, сторінка світло

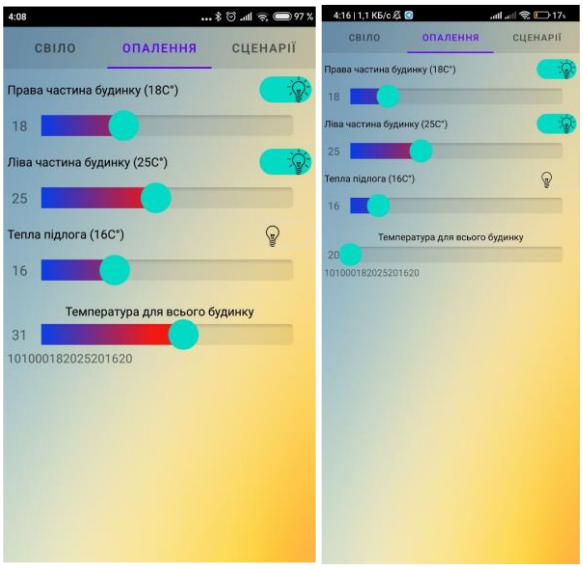


Рис 2.17 Інтерфейс користувача, сторінка опалення

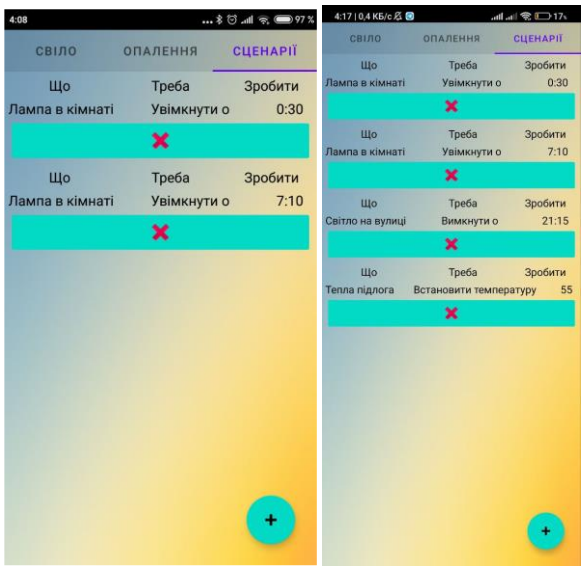


Рис 2.18 Інтерфейс користувача, сторінка сценарії

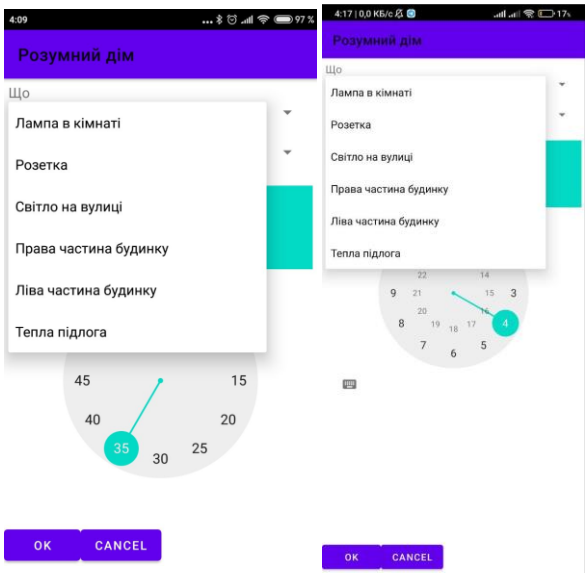


Рис 2.19 Інтерфейс користувача, сторінка додавання сценарію

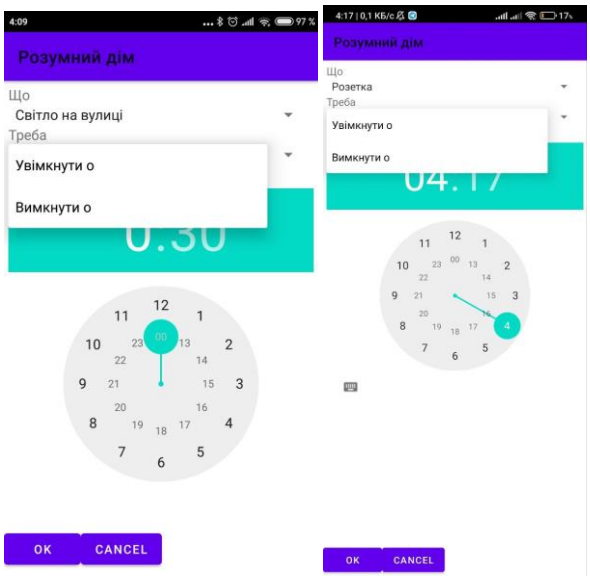


Рис 2.20 Інтерфейс користувача, сторінка додавання сценарію

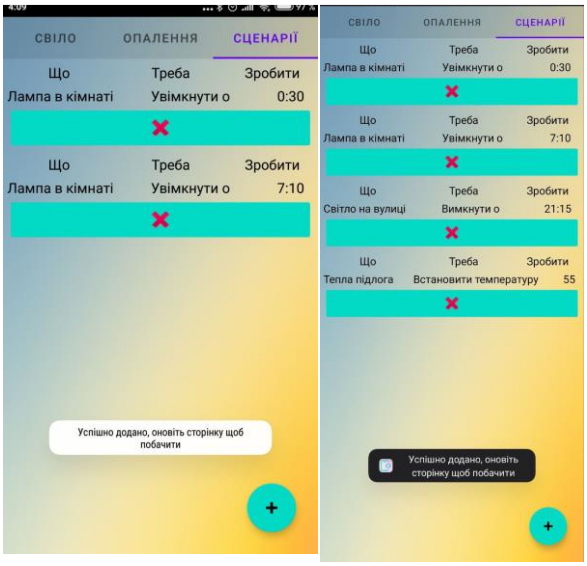


Рис 2.21 Повідомлення про успішне додавання сценарію

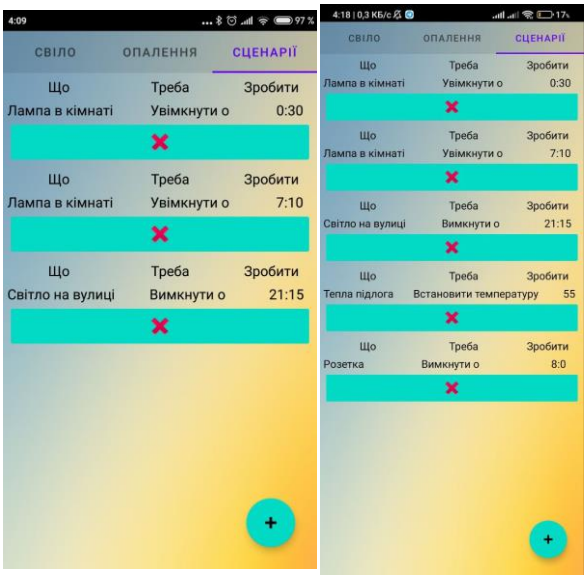


Рис 2.22 Інтерфейс користувача, сторінка сценарії, успішне додавання сценарію

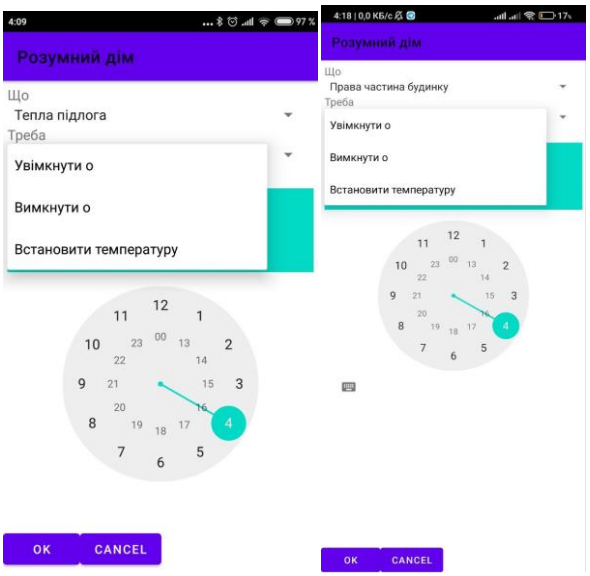


Рис 2.23 Інтерфейс користувача, сторінка додавання сценарію

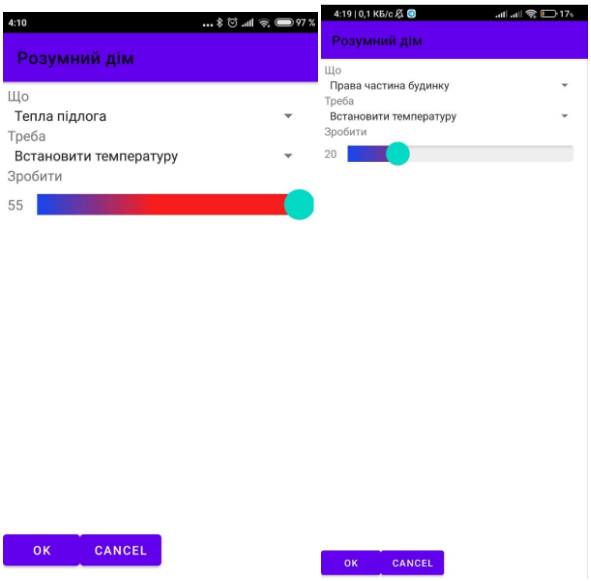


Рис 2.24 Інтерфейс користувача, сторінка додавання сценарію

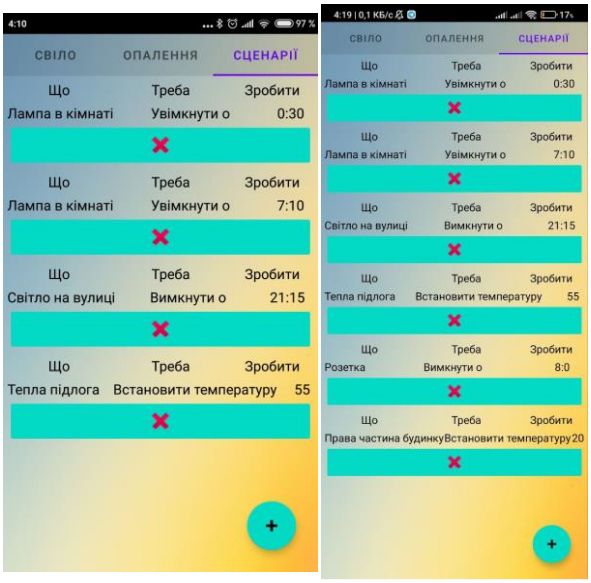


Рис 2.25 Інтерфейс користувача, сторінка сценарії, успішне додавання сценарію

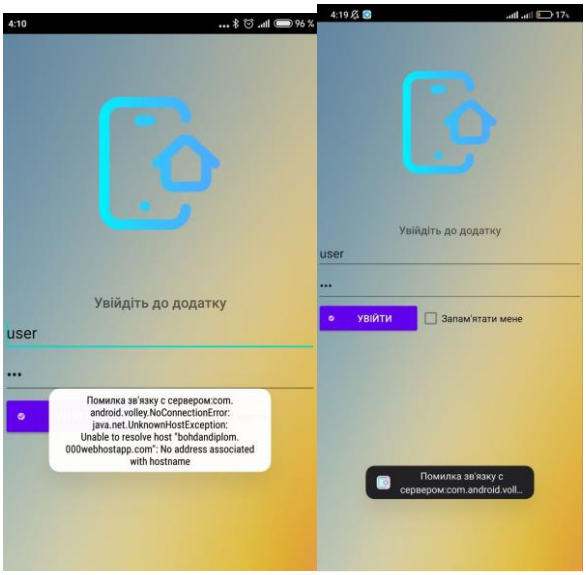


Рис 2.26 Помилка при відсутності інтернет зв’язку на сторінці логіну

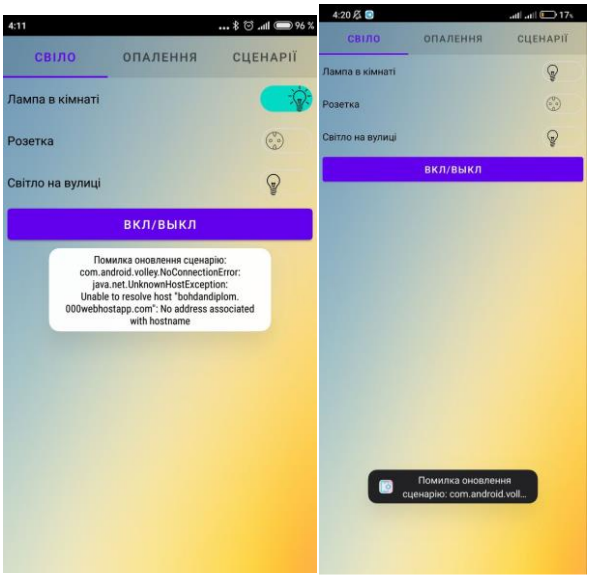


Рис 2.27 Помилка при відсутності інтернет зв’язку під час оновленн я сценарію

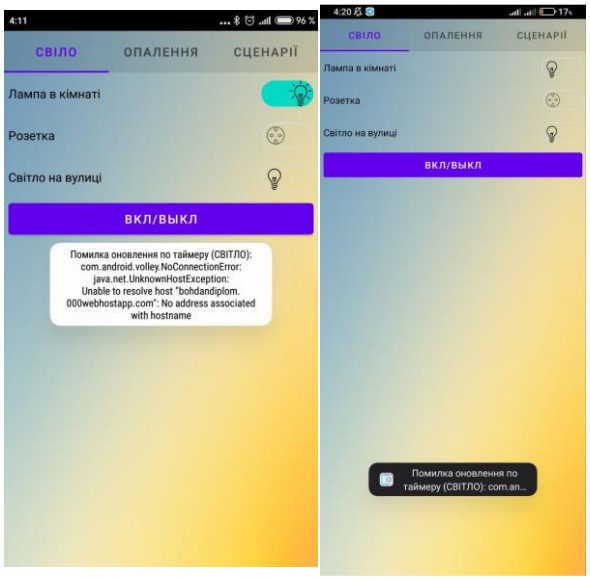


Рис 2.28 Помилка при відсутності інтернет зв’язку на сторінці світло

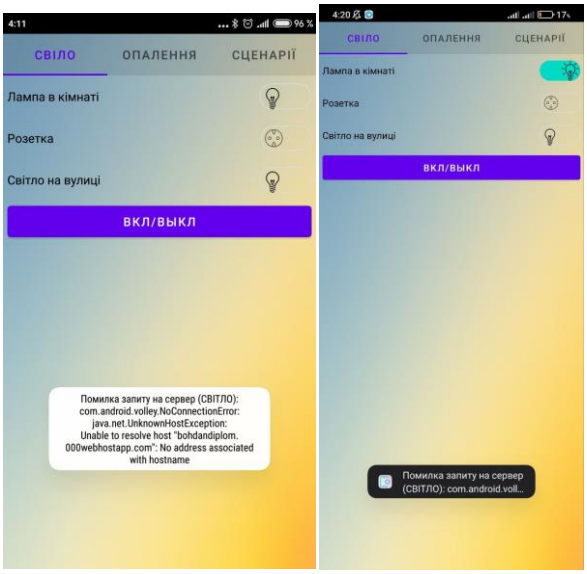


Рис 2.29 Помилка при відсутності інтернет зв’язку на сторінці світло

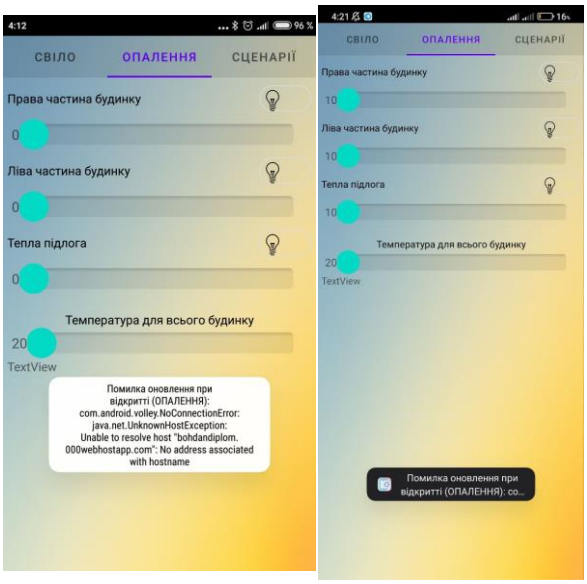


Рис 2.30 Помилка при відсутності інтернет зв’язку на сторінці опалення

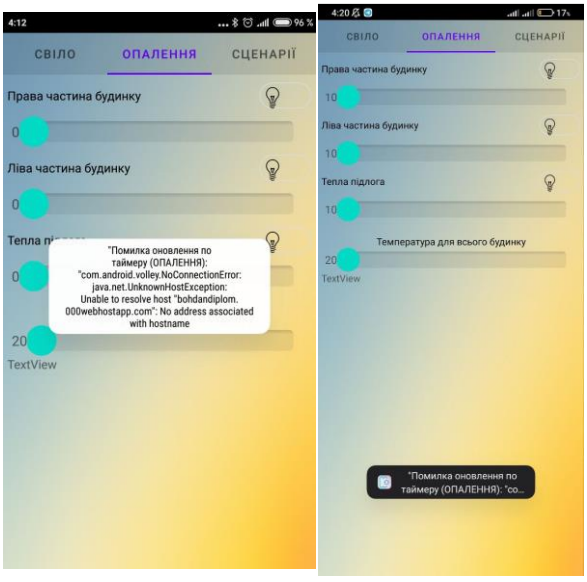


Рис 2.31 Помилка при відсутності інтернет зв’язку на сторінці опалення

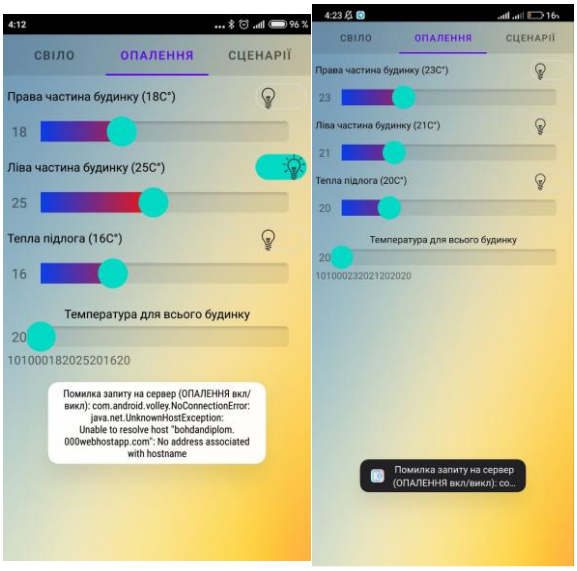


Рис 2.32 Помилка при відсутності інтернет зв’язку на сторінці опалення

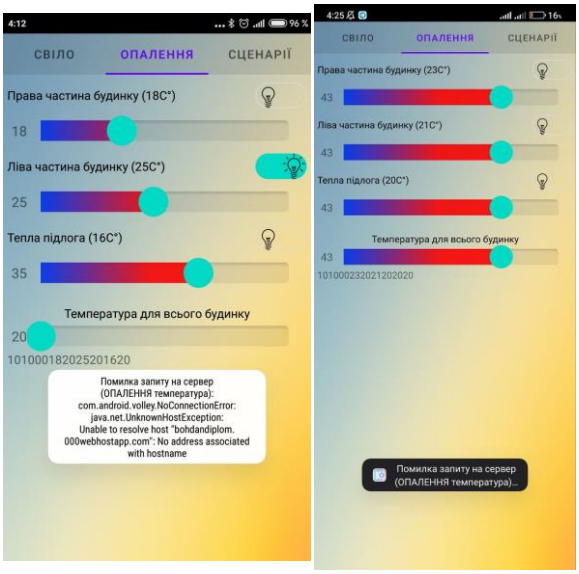


Рис 2.33 Помилка при відсутності інтернет зв’язку на сторінці опалення

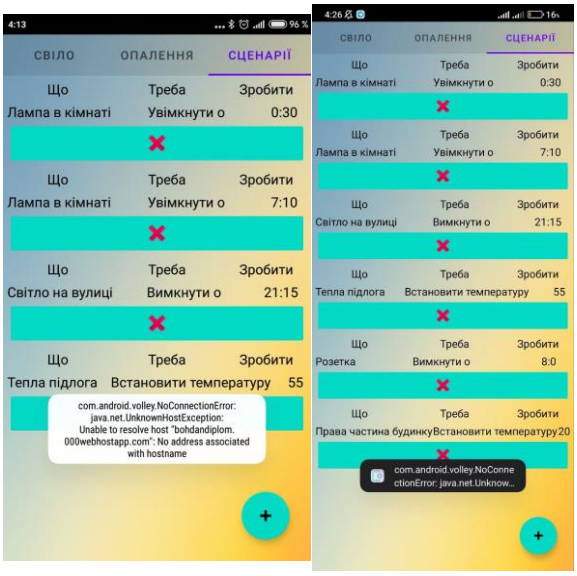


Рис 2.34 Помилка при відсутності інтернет зв’язку на сторінці сценаріїв

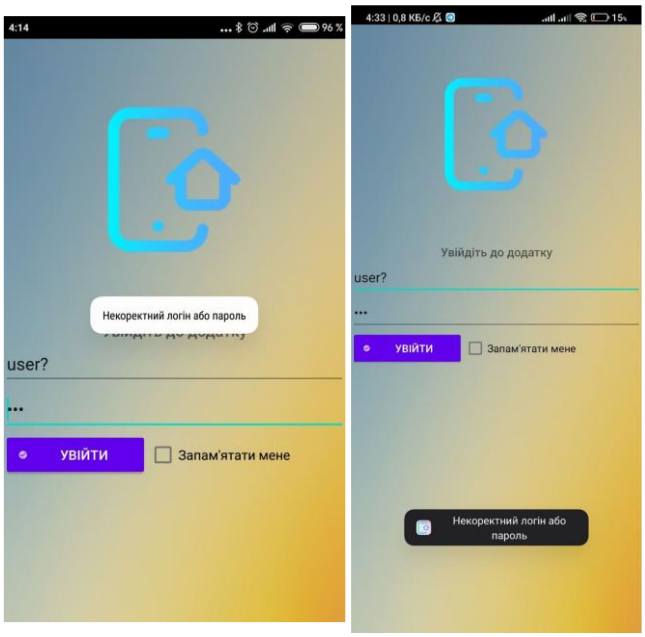


Рис 2.35 Результат введення невірного логіна або паролю

Повідомлення про нештатні ситуації виникають при відсутності інтернет з'єднання і мають наступний текст:

- Помилка зв'язку з сервером (виникає на сторінці логіну рис 2.26);
- Помилка оновлення сценарію (виникає на будь-якій сторінці при оновлені сценарію по таймеру рис 2.27);
- Помилка оновлення по таймеру (СВІТЛО) (виникає на сторінці світло при неможливості оновити дані по таймеру по причині відсутності з'єднання з мережею рис 2.28);
- Помилка запиту на сервер (СВІТЛО) (виникає на сторінці світло при спробі змінити стан елементу рис. 2.29);
- Помилка оновлення при відкритті (ОПАЛЕННЯ) (тригером оновлення при відкритті виступає частина життєвого циклу додатку onResume(), виникає при відкритті сторінки, через неможливість зв'язатися з сервером рис 2.30);
- Помилка оновлення по таймеру (ОПАЛЕННЯ) (виникає на сторінці опалення при неможливості оновити дані по таймеру по причині відсутності з'єднання з мережею рис 2.31);

- Помилка запиту на сервер (ОПАЛЕННЯ вкл/викл) (виникає на сторінці опалення при спробі змінити стан елементу рис. 2.32);
- Помилка запиту на сервер (ОПАЛЕННЯ температура) (виникає на сторінці опалення при спробі змінити темперауру елементу рис. 2.33);
- com.andriod.volley.NoConnectionError (виникає на сторінці сценарії при спробі оновити елементи, якщо вже в масиві даних знаходяться елементи, при відсутності даних в масиві помилка не виникає рис. 2.34);
- Некоректний логін або пароль (виникає при наявності відключення до мережі при некоректному вводу логіну або пароля рис. 2.35);

Порядок роботи програми:

- по перше треба під’єднати всі піни описані в програмі до відповідних роз’ємів плати мікропроцесора Arduino MEGA 2560
- під’єднати шилд Ethernet Shield W5100 Arduino в порт Ethernet під’єднати інтернет шнур підключений до маршрутизатора
- подати живлення на плату
- загрузити мобільний додаток та встановити його на пристрій
- запустити мобільний додаток

Розділ 3
Економічний розділ

3.1. Визначення трудомісткості розробки програмного забезпечення

Початкові дані:

1. передбачуване число операторів програми (Q) – 800;

2. коефіцієнт складності програми (C) – 1,4;

3. коефіцієнт корекції програми в ході її розробки (p) – 0,05;

4. годинна заробітна плата програміста ($C_{ПР}$)– 55 грн/год;

5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі (B) – 1,2;

6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності (K)– 1,2;

7. вартість машино-години ЕОМ ($З_{МВ}$) – 15 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_д \text{, людино-годин,} \qquad (3.1)$$

де t_o – витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ – витрати праці на налагодження програми на ЕОМ;

$t_д$ – витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p), \qquad (3.2)$$

де q – передбачуване число операторів;

C – коефіцієнт складності програми;

p – коефіцієнт кореляції програми в ході її розробки.

$$Q = 800 \cdot\ 1{,}4\ \cdot (1 - 0{,}05) = 1064$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \ldots 85) \cdot K}, \text{ людино-годин} \qquad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

K – коефіцієнт кваліфікації програміста, обумовлений стажем роботи з даної спеціальності;

$$t_u = \frac{1064 \cdot 1{,}2}{85 \cdot 1{,}2} = 12{,}51 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20 \ldots 25) \cdot K}; \qquad (3.4)$$

$$t_a = \frac{1064}{23 \cdot 1{,}2} = 38{,}55 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \ldots 25) \cdot K}; \qquad (3.5)$$

$$t_n = \frac{1064}{21 \cdot 1{,}2} = 42{,}22 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{\text{отл}} = \frac{Q}{(4...5) \cdot K}; \quad (3.6)$$

$$t_n = \frac{1064}{4 \cdot 1{,}2} = 221{,}67 \text{ людино-годин},$$

– за умови комплексного налагодження завдання:

$$t_{\text{отл}}^{\text{к}} = 1{,}2 \cdot t_{\text{отл}}; \quad (3.7)$$

$$t_{\text{отл}}^{\text{к}} = 1{,}2 \cdot 221{,}67 = 266{,}04 \text{ людино-годин}$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o}; \quad (3.8)$$

де $t_{\partial p}$ – трудомісткість підготовки матеріалів і рукопису

$$t_{\partial} = \frac{Q}{(15...20) \cdot K}; \quad (3.9)$$

$$t_{\partial p} = \frac{1064}{20 \cdot 1{,}2} = 44{,}33 \text{ людино-годин.}$$

$t_{\partial o}$– трудомісткість редагування, печатки й оформлення документації

$$t_{\partial o} = 0{,}75 \cdot t_{\partial \text{р}}; \quad (3.10)$$

$$t_{\partial o} = 0{,}75 \cdot 44{,}33 = 33{,}24 \text{ людино-годин.}$$

$$t_д = 44{,}33 + 33{,}24 = 77{,}57 \text{ людино-годин.}$$

Отримаємо трудомісткість розробки програмного забезпечення:

t = 50 + 12,51 + 38,55 + 42,22 + 266,04 + 77,57 = 486,89 людино-годин.

У результаті ми розрахували, що в загальній складності необхідно близько 486,89 людино-годин для розробки даного програмного забезпечення.

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ включають витрати на заробітну плату виконавця програми З/п і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{ПО} = З_{ЗП} + З_{МВ}, \text{ грн,} \qquad (3.11)$$

де $З_{ЗП}$ – заробітна плата виконавців, яка визначається за формулою:

$$З_{ЗП} = t \cdot C_{ПР}, \text{ грн,} \qquad (3.12)$$

де t – загальна трудомісткість, людино-годин;

$C_{ПР}$ – середня годинна заробітна плата програміста, грн/година

$$З_{зп} = 486{,}89 \cdot 55 = 26778{,}95 \text{ грн.}$$

$З_{МВ}$ – Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{МВ} = t_{отл} \cdot C_{М}, \text{ грн,} \qquad (3.13)$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ, год.

$C_{МЧ}$ – вартість машино-години ЕОМ, грн/год.

$$З_{МВ} = 266{,}04 \cdot 15 = 3990{,}6 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$К_{ПО} = 26778{,}95 + 3990{,}6 = 30769{,}55 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ мес.} \qquad (3.14)$$

де B_k – число виконавців;

F_p – місячний фонд робочого часу (при 40 годинному робочому тижні Fр=176 годин).

Витрати на створення программного продукту:

$$T = \frac{486{,}89}{1 \cdot 176} = 2{,}8 \text{ міс.}$$

Висновки. Інформаційна система має вартість 30769,5 грн. Ймовірний очікуваний час розробки – 2,8 місяці при стандартному 40-годинному робочому тижні і 178-годинному робочому місяці. Цей термін пов’язаний з кількістю операторів і включає в себе час для дослідження та розробку алгоритму розв’язання задачі, розробку дизайну, створення документації, розробку баз даних і програмування контролеру. На розробку інформаційної системи буде витрачено 486,89 людино-годин.

ВИСНОВКИ

Створення інформаційної системи «розумний дім» актуально сьогодні, як ніколи. Інтернет речей набуває популярності щодня. Метою кваліфікаційної роботи є створення інформаційної системи «Розумний дім» на базі мікропроцесору Arduino. В процесі написання кваліфікаційної роботи було встановлено, що дана інформаційна система корисна для оптимізації проживання в будинку і для полегшення управління процесами які проходять в будинку.

Інформаційна система має наступний функціонал. Програма для контролера на базі мікропроцесора Arduino виконує такі функції:

- зчитування даних з периферії;
- зберігання інформації про периферію;
- зміна цільового стану елементів периферії з аналогового вводу за допомогою кнопок;
- зміна цільового стану елементів периферії по отриманих даним зі серверу;
- обробка і розподілення отриманих даних по змінним в програмі;
- відправка оброблених даних на сервер;
- отримання даних про периферію з сервера;

Сервер містить:

- SQL базу даних з таблицями розподілених по категоріям відповідно до різних типів периферії контролеру;
- таблицю де зберігається інформація про стан увімкнутий чи вимкнутий той ти інший елемент периферії;
- таблицю де зберігаються дані про поточну і цільову температуру в різних частинах будинку;
- таблицю з описом сценарію роботи елементів периферії контролеру;

Мобільний додаток на базі ОС Android 7.0 або вище містить наступні функції:

– авторизування користувача;
– відображення стану освітлювальних приладів;
– відображення поточної температури різних частин будинку;
– можливість встановлення цільової температури для різних частин будинку;
– відображення встановлених сценаріїв роботи;
– створення сценаріїв роботи по шаблону;
– видалення сценаріїв;

Поставленні задачі в ході виконання кваліфікаційної роботи виконані в повному об’ємі за допомогою серед розробки Android Studio, Arduino IDE та PhpMyAdmin. Мобільний додаток працює під керуванням Android, яка широко використовується цільовою аудиторією продукту. Додаток реалізований на мові програмування Java з використанням набору засобів розробки програмного забезпечення (SDK) –Volley.

Актуальність поставленої задачі обумовлюється широким попитом на такі програмні продукти. На сьогоднішній день мобільна розробка є однією з найбільш популярних напрямків в інформаційних технологіях. Крім того, розумний дім - тренд останніх років, який тільки підвищує інтерес до даного проекту.

В «Економічному розділі» визначено трудомісткість розробки програмного забезпечення (486,89 чол-год), підраховані витрати на створення програмного забезпечення (30769,5грн.) і гаданий період розробки (2,8 міс.).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Как задать ID динамически созданному View. URL: https://ru.stackoverflow.com/questions/437897/%d0%9a%d0%b0%d0%ba-%d0%b7%d0%b0%d0%b4%d0%b0%d1%82%d1%8c-id-%d0%b4%d0%b8%d0%bd%d0%b0%d0%bc%d0%b8%d1%87%d0%b5%d1%81%d0%ba%d0%b8%d1%81%d0%be%d0%b7%d0%b4%d0%b0%d0%bd%d0%bd%d0%be%d0%bc%d1%83-view
2. Что такое PHP? URL: https://www.php.net/manual/ru/intro-whatis.php
3. Возможности PHP. URL: https://www.php.net/manual/ru/intro-hatcando.php
4. Arduino Mega 2560: распиновка, схема подключения и рограммирование. URL:http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D%BA%D1%82%D1%8B:arduino-mega-2560
5. Программирование Ардуино. URL: https://doc.arduino.ua/ru/prog/
6. AVR Libc. URL: http://www.nongnu.org/avr-libc/user-manual/index.html
7. Выбор элемента в ListView. URL: https://metanit.com/java/android/5.2.php
8. как избежать переполнение массива ардуино. URL:https://qna.habr.com/q/432903
9. Как сделать HTTP-запрос или HTTPS-запрос в PHP скрипте. URL: https://fructcode.com/ru/blog/how-to-get-http-site-with-php/
10. Умный дом приложение – революция. URL: https://www.fibaro.com/ru/smart-home-app/
11. Android: Потоки. URL: http://developer.alexanderklimov.ru/android/theory/thread.php
12. Volley overview. URL: https://google.github.io/volley/
13. Центр справки Java. URL: https://www.java.com/ru/download/help/index.html
14. Android Development. URL: https://developer.android.com/
15. Android Manifest: Specifying Android App and SDK Versions. URL: https://howtodoinjava.com/android/android-manifest-specifying-android-app-and-sdk-versions/

16. Custom buttons for android studio. URL: https://www.codegrepper.com/code-examples/java/custom+buttons+for+android+studio

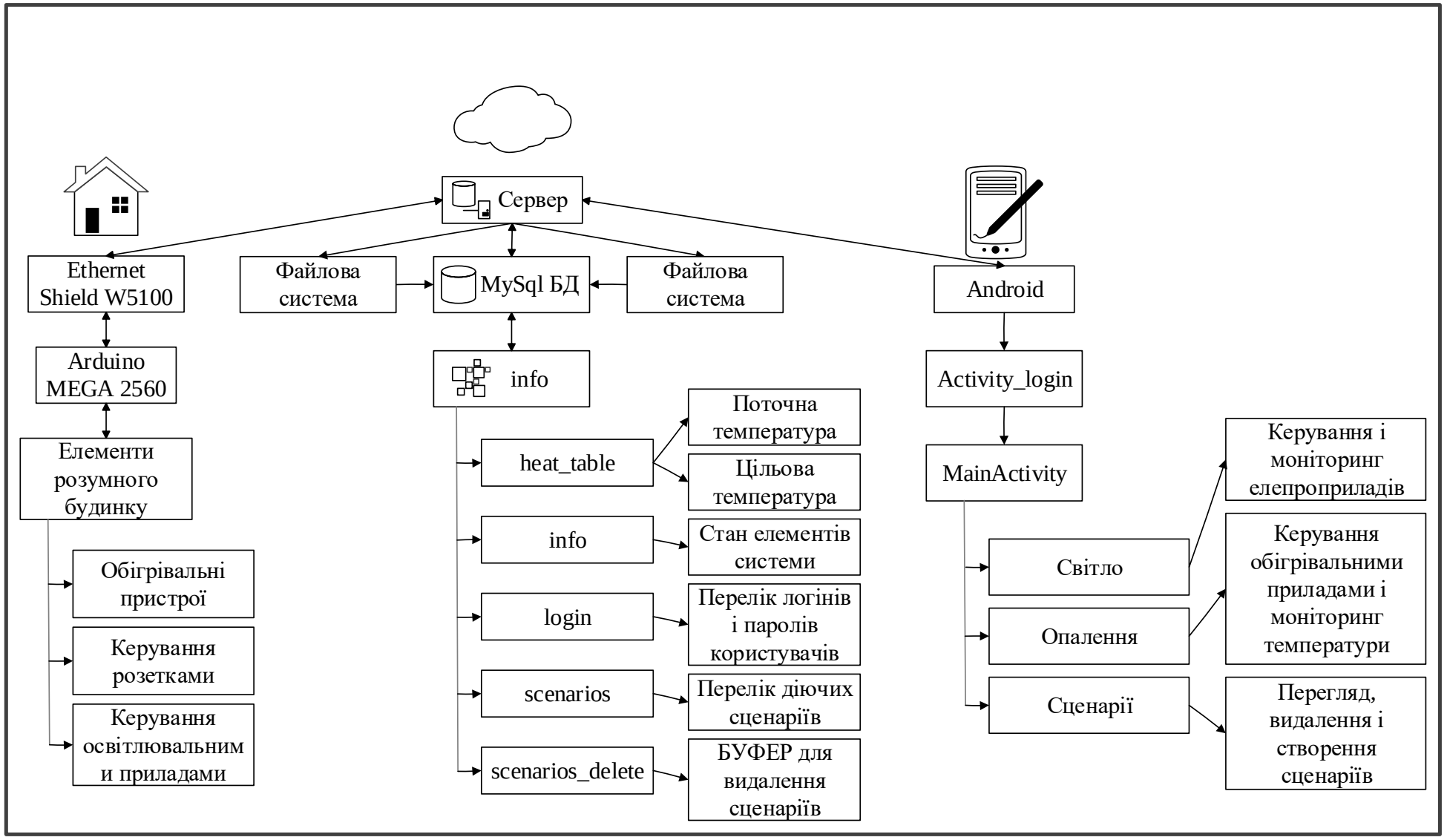
17. Custom buttons in Android. URL:https://www.codebrainer.com/blog/13-designs-for-buttons-every-android-beginner-should-know

18. Custom seekbar (thumb size, color and background) . URL: https://stackoverflow.com/questions/41693154/custom-seekbar-thumb-size-color-and-background

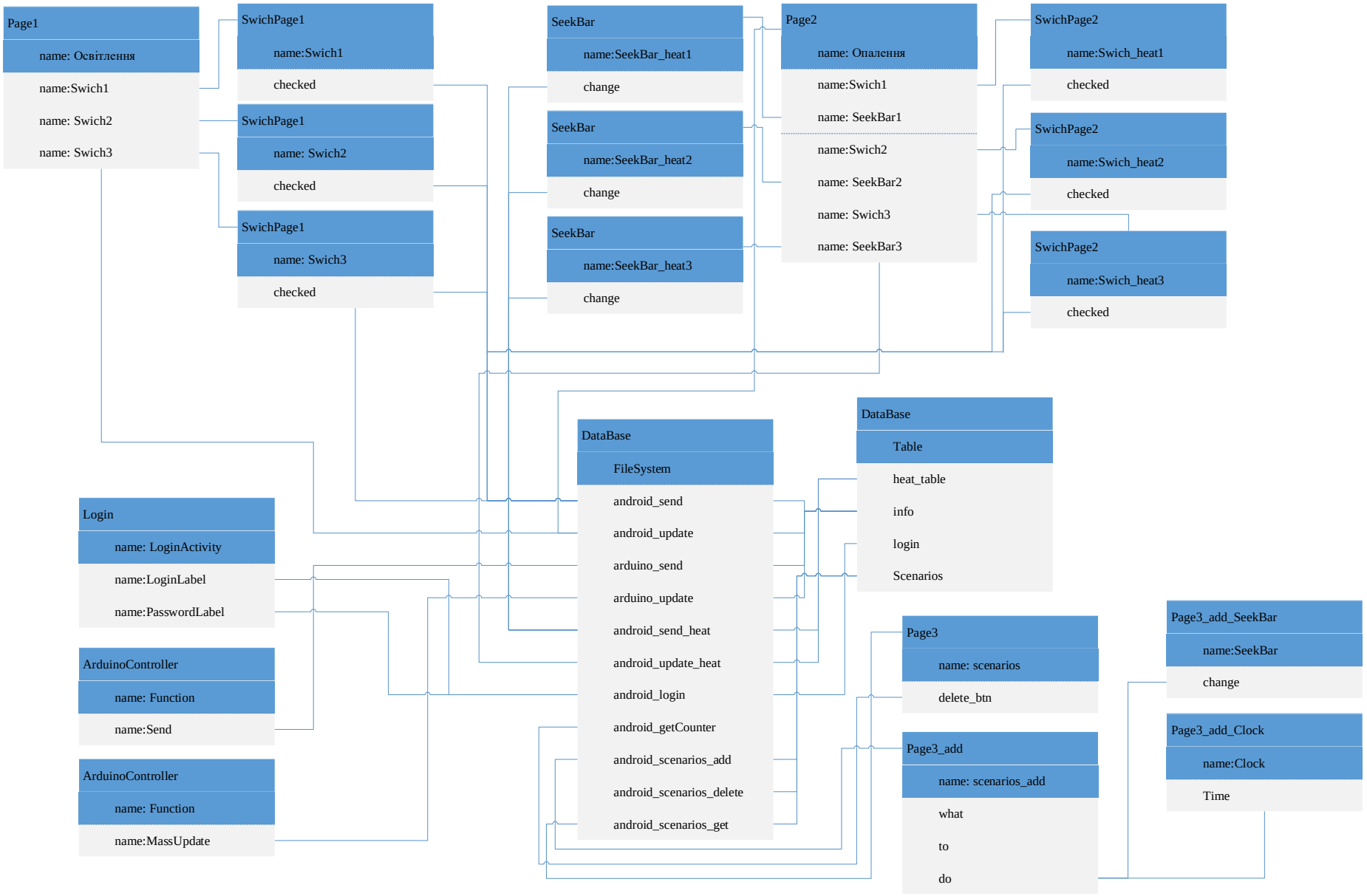
19. How to custom switch button? . URL: https://stackoverflow.com/questions/23358822/how-to-custom-switch-button

20. 000webhost. URL: https://www.000webhost.com/members/website/bohdandiplom/database

ДОДАТОК А


Сервер
Ethernet Shield W5100
Файлова система
MySql БД
Файлова система
Android
Arduino MEGA 2560
info
Activity_login
Елементи розумного будинку
heat_table
Поточна температура
Цільова температура
MainActivity
Обігрівальні пристрої
info
Стан елементів системи
Світло
Керування і моніторинг елепроприладів
Керування розетками
login
Перелік логінів і паролів користувачів
Опалення
Керування обігрівальними приладами і моніторинг температури
Керування освітлювальним и приладами
scenarios
Перелік діючих сценаріїв
Сценарії
Перегляд, видалення і створення сценаріїв
scenarios_delete
БУФЕР для видалення сценаріїв

ДОДАТОК Б


Page1
name: Освітлення
name:Swich1
name: Swich2
name: Swich3
SwichPage1
name:Swich1
checked
SwichPage1
name: Swich2
checked
SwichPage1
name: Swich3
checked
SeekBar
name:SeekBar_heat1
change
SeekBar
name:SeekBar_heat2
change
SeekBar
name:SeekBar_heat3
change
Page2
name: Опалення
name:Swich1
name: SeekBar1
name:Swich2
name: SeekBar2
name: Swich3
name: SeekBar3
SwichPage2
name:Swich_heat1
checked
SwichPage2
name:Swich_heat2
checked
SwichPage2
name:Swich_heat3
checked
DataBase
FileSystem
android_send
android_update
arduino_send
arduino_update
android_send_heat
android_update_heat
android_login
android_getCounter
android_scenarios_add
android_scenarios_delete
android_scenarios_get
DataBase
Table
heat_table
info
login
Scenarios
Login
name: LoginActivity
name:LoginLabel
name:PasswordLabel
ArduinoController
name: Function
name:Send
ArduinoController
name: Function
name:MassUpdate
Page3
name: scenarios
delete_btn
Page3_add
name: scenarios_add
what
to
do
Page3_add_SeekBar
name:SeekBar
change
Page3_add_Clock
name:Clock
Time

ДОДАТОК В

КОД ПРОГРАМИ

```
#include <SPI.h> //код контролеру C++
#include <Ethernet.h>
IPAddress ip(192, 168, 1, 8);
byte mac[] = {  0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02};
char destination[] = "www.bohdandiplom.000webhostapp.com";
EthernetClient client;
//Все переменные светодиодов
int led1;
int led2;
int led3;
//Весь БУФФЕР для данных
struct Mass {
  int led1;
  int led2;
  int led3;
  int flag1;
  int flag2;
  int flag3;
};
unsigned long previousMillis = 0;
const int updateInterval = 10000;

void setup()
{

  Serial.begin(115200);
  //Лампочки
  pinMode(44, OUTPUT);//
  pinMode(45, OUTPUT);//
  pinMode(46, OUTPUT);//send
  pinMode(47, OUTPUT);//update
  pinMode(48, OUTPUT);
  pinMode(49, OUTPUT);
  pinMode(50, OUTPUT);
  pinMode(51, OUTPUT);
  //Кнопочки
  pinMode(22, INPUT);// SendMass
  pinMode(23, INPUT);//UpdateMass
  pinMode(24, INPUT);
  pinMode(25, INPUT);//
  pinMode(26, INPUT);
  pinMode(27, INPUT);//
  pinMode(28, INPUT);
  pinMode(29, INPUT_PULLUP);//
  Ethernet.begin(mac, ip);
  delay(1000);
  Mass massled;
  massled = updatemass(1, 2, 3);
  led1 = massled.led1;
  led2 = massled.led2;
  led3 = massled.led3;
  if ( led1 == 1 )  digitalWrite(44, 1); else if ( led1 == 0 ) digitalWrite(44, 0);
  if ( led2 == 1 )  digitalWrite(45, 1); else if ( led2 == 0 ) digitalWrite(45, 0);
  if ( led3 == 1 )  digitalWrite(46, 1); else if ( led3 == 0 ) digitalWrite(46, 0);
  Serial.println("READY");
}

boolean flag = 0;
boolean flag2 = 0;
boolean flag3 = 0;
```

```
void loop()
{
 Mass massled;

 delay(5000);

 // Send
 int buttoninf22 = digitalRead(22);
 if (buttoninf22 == HIGH) {
  digitalWrite(47, 1);
  Serial.println("SendMass");
  sendmass(1, led1, flag, 2, led2, flag2, 3, led3, flag3);
 } else digitalWrite(47, 0);

 // Update
 unsigned long currentMillis = millis();
 if ( currentMillis - previousMillis >= updateInterval) {
  previousMillis = currentMillis;
  digitalWrite(47, 1);
  Serial.println("UpdateMass");
  massled = updatemass(1, 2, 3);
  led1 = massled.led1;
  led2 = massled.led2;
  led3 = massled.led3;
  if ( led1 == 1 )  digitalWrite(44, 1); else if ( led1 == 0 ) digitalWrite(44, 0);
  if ( led2 == 1 )  digitalWrite(45, 1); else if ( led2 == 0 ) digitalWrite(45, 0);
  if ( led3 == 1 )  digitalWrite(46, 1); else if ( led3 == 0 ) digitalWrite(46, 0);
  Serial.println (massled.led1); Serial.println (massled.led2); Serial.println (massled.led3);
  digitalWrite(47, 0);
 }
  //29
 boolean btnState1 = digitalRead(29); //!
 if (btnState1 == 1  && flag == 0) { // Нажатие
  flag = 1;
  Serial.println("press29");
 } if (btnState1 == 0 && flag == 1) { // Отпускание
  flag = 0;
  Serial.println("release29");
  led1 = !led1;
  digitalWrite(44, led1);
  sendpin(1, led1, flag);
 }

 //25
 boolean btnState2 = digitalRead(25);
 if (btnState2 == 1  && flag2 == 0) { // Нажатие
  flag2 = 1;
  Serial.println("press25");
 } if (btnState2 == 0 && flag2 == 1) { // Отпускание
  flag2 = 0;
  Serial.println("release25");
  led2 = !led2;
  digitalWrite(45, led2);
  sendpin(2, led2, flag2);
 }

 //27
 boolean btnState3 = digitalRead(27);
 if (btnState3 == 1  && flag3 == 0) { // Нажатие
  flag3 = 1;
  Serial.println("press27");
 } if (btnState3 == 0 && flag3 == 1) { // Отпускание
```

```
    flag3 = 0;
    Serial.println("release27");
    led3 = !led3;
    digitalWrite(46, led3);
    sendpin(3, led3, flag3);
   }

  int buttoninf23 = digitalRead(23);
    if (buttoninf23 == HIGH) {
    digitalWrite(47, 1);
    Serial.println("UpdateMass");
    massled = updatemass(1, 2, 3);
    led1 = massled.led1;
    led2 = massled.led2;
    led3 = massled.led3;
    if ( led1 == 1 )  digitalWrite(44, 1); else if ( led1 == 0 ) digitalWrite(44, 0);
    if ( led2 == 1 )  digitalWrite(45, 1); else if ( led2 == 0 ) digitalWrite(45, 0);
    if ( led3 == 1 )  digitalWrite(46, 1); else if ( led3 == 0 ) digitalWrite(46, 0);
    Serial.println (massled.led1); Serial.println (massled.led2); Serial.println (massled.led3);
    } else digitalWrite(47, 0);

}

void sendmass(int p, int b, int f, int p2, int b2, int f2, int p3, int b3, int f3) {

  client.connect(destination, 80);
  if (client.connect(destination, 80)) {
    Serial.println("Sendmass TRUE");
    /*Serial.println(b);
      Serial.println(b2);
      Serial.println(b3);
      Serial.print(f);
      Serial.print(f2);
      Serial.println(f3);*/
    client.println("POST /sendmass.php HTTP/1.1");
    client.println("Host: bohdandiplom.000webhostapp.com");
    client.println("User-Agent: arduino-ethernet");
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.println("Content-Length: 74");                                    ////////////// 74//69
    client.println();
    client.print("pin=pin");//7
    client.print(p);
    client.print("&");
    client.print("bool=");
    client.print(b);
    client.print("&");//16
    client.print("flag=");
    client.print(f);
    client.print("&");
    client.print("pin2=pin");
    client.print(p2);
    client.print("&");//33
    client.print("bool2=");
    client.print(b2);
    client.print("&");
    client.print("flag2=");
    client.print(f2);
    client.print("&");//49
    client.print("pin3=pin");
    client.print(p3);
    client.print("&");
    client.print("bool3=");
    client.print(b3);
```

```
    client.print("&");
    client.print("flag3=");
    client.print(f3);
    client.println();
    client.println("Connection: close");
   }
   else
    Serial.println("Sendmass FALSE");

   if (client.connected()) {
    client.stop();
    client.flush();
   }
}

Mass updatemass(int p, int p2, int p3) {
   Mass massled;
   EthernetClient client1;
   client1.connect(destination, 80);
   if (client1.connect(destination, 80)) {
    Serial.println("Update TRUE ");
    client1.println("POST /php.php HTTP/1.1");
    client1.println("Host: bohdandiplom.000webhostapp.com");
    client1.println("User-Agent: arduino-ethernet");
    client1.println("Content-Type: application/x-www-form-urlencoded");
    client1.println("Content-Length: 28");
    client1.println();
    client1.print("pin=pin");
    client1.print(p);
    client1.print("&");
    client1.print("pin2=pin");
    client1.print(p2);
    client1.print("&");
    client1.print("pin3=pin");
    client1.print(p3);
    client1.println();
    //Прием данных
    Serial.println("Update...");
    String c = client1.readString();
    //Serial.println(c);
    char arr[c.length() + 1];
    c.toCharArray(arr, c.length() + 1);
    /*Serial.print ("Boolean = ");
      Serial.println (arr[278]); //boolean
      Serial.print ("Boolean2 = ");
      Serial.println (arr[304]); //boolean
      Serial.print ("Boolean3 = ");
      Serial.println (arr[330]); //boolean*/
    massled.led1 = arr[278] - '0';
    massled.led2 = arr[304] - '0';
    massled.led3 = arr[330] - '0';

    Serial.println (massled.led1); Serial.println (massled.led2); Serial.println (massled.led3);
    client1.println("Connection: close");
    Serial.println("Update done; connection: close");
   }
   else
    Serial.println("FALSE update");
   if (client1.connected()) {
    client1.stop();
    client1.flush();
   }
```

```
  return massled;
}

void sendpin(int p, int b, int f) {
  Serial.print(p);
  Serial.print(b);
  Serial.print(f);
  client.connect(destination, 80);
  if (client.connect(destination, 80)) {
    Serial.println("Sendpin TRUE");
    client.println("POST /arduino_send.php HTTP/1.1");
    client.println("Host: bohdandiplom.000webhostapp.com");
    client.println("User-Agent: arduino-ethernet");
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.println("Content-Length: 23");                              //////////// 74//69
    client.println();
    client.print("pin=pin");//7
    client.print(p);
    client.print("&");
    client.print("bool=");
    client.print(b);
    client.print("&");//16
    client.print("flag=");
    client.print(f);
    client.println();
    client.println("Connection: close");
  }
  else
    Serial.println("Sendpin FALSE");

  if (client.connected()) {
    client.stop();
    client.flush();
  }
}
//android_GetCounter
<?php
$servername = "localhost";
$database = "id18087376_info";
$username = "id18087376_arduino";
$password = "kv~-tt[V7WS>)AfF";
$conn = new mysqli($servername, $username, $password, $database);
header('Content-Type:text/plain;charset=utf-8');
function C($url){
 return json_decode(file_get_contents($url));
}
$conn = new mysqli($servername, $username, $password, $database);
if($conn->connect_error){
    die("Ошибка: " . $conn->connect_error);
}
//SELECT MAX(counter) FROM scenarios
$sql = "SELECT * FROM scenarios ORDER BY counter DESC LIMIT 1";

    if($result = $conn->query($sql)){
    foreach($result as $row){
        echo $row["counter"] ;
        }
    } else{echo "Ошибка: " . $conn->error;}

$conn->close();
?>
```

PAGE2.JAVA

```
package com.example.myapplication;
import android.annotation.SuppressLint;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.widget.SwitchCompat;
import androidx.fragment.app.Fragment;
import android.text.Html;
import android.text.Spanned;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import android.widget.CompoundButton;
import android.widget.SeekBar;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import java.util.HashMap;
import java.util.Map;
import java.util.Timer;
import java.util.TimerTask;

public class page2 extends Fragment {

    private SwitchCompat swich1;
    private SwitchCompat swich2;
    private SwitchCompat swich3;


    private SeekBar seekBar1;
    private TextView t1;
    private SeekBar seekBar2;
    private TextView t2;
    private SeekBar seekBar3;
    private TextView t3;
    private char ic_temp = 0x00B0;
    private SeekBar seekBar_all;
    private TextView t_all;
    private String temp1_1;
    private String temp1_2;
    private String temp2_1;
    private String temp2_2;
    private String temp3_1;
    private String temp3_2;
    private TextView textView;

    private RequestQueue mRequestQueue;
    private StringRequest mStringRequest;
    private Timer timer;
    private TimerTask updateTimerTask;
```

```
    public page2() {
        // Required empty public constructor
    }

    public static page2 newInstance(String param1, String param2) {
        page2 fragment = new page2();

        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_page2, container, false);
    }

    //PINS
    @Override
    public void onResume() {
        super.onResume();
        t1.setText(String.valueOf(seekBar1.getProgress()));
        t2.setText(String.valueOf(seekBar2.getProgress()));
        t3.setText(String.valueOf(seekBar3.getProgress()));
        new Thread(new Runnable() {
            @Override
            public void run() {

                RequestQueue requestQueue = Volley.newRequestQueue(getActivity().getApplicationContext());
                StringRequest stringRequest = new StringRequest(Request.Method.POST,
"https://bohdandiplom.000webhostapp.com/android_update_heat.php", new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        String ansver = response;
                        int i,b,f;
                        ansver = response;

                        if((ansver.charAt(0))%2==0){
                            swich1.setChecked(false);
                        } else swich1.setChecked(true);

                        if((ansver.charAt(2))%2==0){
                            swich2.setChecked(false);
                        } else swich2.setChecked(true);

                        if((ansver.charAt(4))%2==0){
                            swich3.setChecked(false);
                        } else swich3.setChecked(true);
                        temp1_1 = String.valueOf((ansver.charAt(6)));//Target temp, not current now (+2)
                        temp1_2 = String.valueOf((ansver.charAt(7)));//Target temp, not current now (+2)
                        temp2_1 = String.valueOf((ansver.charAt(10)));//Target temp, not current now (+2)
                        temp2_2 = String.valueOf((ansver.charAt(11)));//Target temp, not current now (+2)
```

```
                temp3_1 = String.valueOf((ansver.charAt(14)));//Target temp, not current now (+2)
                temp3_2 = String.valueOf((ansver.charAt(15)));//Target temp, not current now (+2)
                swich1.setText( "Права частина будинку (" + temp1_1 + temp1_2 + "C" + ic_temp + ")");
                swich2.setText( "Ліва частина будинку (" + temp2_1 + temp2_2 + "C" + ic_temp + ")");
                swich3.setText( "Тепла підлога (" + temp3_1 + temp3_2 + "C" + ic_temp + ")");
                seekBar1.setProgress(Integer.parseInt(temp1_1+temp1_2));
                seekBar2.setProgress(Integer.parseInt(temp2_1 + temp2_2));
                seekBar3.setProgress(Integer.parseInt(temp3_1 + temp3_2));
                textView.setText(ansver);
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast toast = Toast.makeText(getActivity().getApplicationContext(),
                        "Помилка оновлення при відкритті (ОПАЛЕННЯ): "+ error.toString(), Toast.LENGTH_LONG);
                toast.show();

            }
        }){
            @Override
            public Map<String, String> getParams() {
                Map<String, String> map = new HashMap<>();
                map.put("pin", "pin4" );
                map.put("pin2", "pin5" );
                map.put("pin3", "pin6" );
                map.put("heat", "1" );
                map.put("heat2", "2" );
                map.put("heat3", "3" );
                return map;
            }
        };
        requestQueue.add(stringRequest);
      }
   }).start();

   timer = new Timer();
   updateTimerTask = new page2.updateTimer();
   timer.schedule(updateTimerTask, 2500, 3500);
}
//SeekBar
@Override
public void onPause(){
   super.onPause();
   timer.cancel();
}

//PINS
@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {

   swich1 = view.findViewById(R.id.switch1);
   swich2 = view.findViewById(R.id.switch2);
   swich3 = view.findViewById(R.id.switch3);
   seekBar1 = view.findViewById(R.id.seekBar1);
   seekBar2 = view.findViewById(R.id.seekBar2);
   seekBar3 = view.findViewById(R.id.seekBar3);
   seekBar_all = view.findViewById(R.id.seekBar_all);
   t1 = view.findViewById(R.id.textView1);
```

```
t2 = view.findViewById(R.id.textView2);
t3 = view.findViewById(R.id.textView3);
t_all = view.findViewById(R.id.textView_all);
seekBar1.setMax(55);
seekBar1.setSplitTrack(false);
seekBar2.setMax(55);
seekBar3.setMax(55);

textView = view.findViewById(R.id.textView4);
t1.setText(String.valueOf(seekBar1.getProgress()));
t2.setText(String.valueOf(seekBar2.getProgress()));
t3.setText(String.valueOf(seekBar3.getProgress()));

seekBar1.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        t1.setText(String.valueOf(seekBar1.getProgress()));
        if(seekBar1.getProgress()<10)
        {
            seekBar1.setProgress(10);
        }
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
        t1.setText(String.valueOf(seekBar1.getProgress()));
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        t1.setText(String.valueOf(seekBar1.getProgress()));
        Responce_heat("1", Integer.valueOf(seekBar1.getProgress()));
    }
});
seekBar2.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        t2.setText(String.valueOf(seekBar2.getProgress()));
        if(seekBar2.getProgress()<10)
        {
            seekBar2.setProgress(10);
        }
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
        t2.setText(String.valueOf(seekBar2.getProgress()));
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        t2.setText(String.valueOf(seekBar2.getProgress()));
        Responce_heat("2", Integer.valueOf(seekBar2.getProgress()));
    }
});

seekBar3.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
```

```
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        t3.setText(String.valueOf(seekBar3.getProgress()));
        if(seekBar3.getProgress()<10)
        {
            seekBar3.setProgress(10);
        }
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
        t3.setText(String.valueOf(seekBar3.getProgress()));
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        t3.setText(String.valueOf(seekBar3.getProgress()));
        Responce_heat("3", Integer.valueOf(seekBar3.getProgress()));
    }
});

seekBar_all.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        if(seekBar_all.getProgress()<10)
        {
            seekBar_all.setProgress(10);
        }
        seekBar1.setProgress(seekBar_all.getProgress());
        seekBar2.setProgress(seekBar_all.getProgress());
        seekBar3.setProgress(seekBar_all.getProgress());
        t_all.setText(String.valueOf(seekBar_all.getProgress()));
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        Responce_heat("1", Integer.valueOf(seekBar_all.getProgress()));
        Responce_heat("2", Integer.valueOf(seekBar_all.getProgress()));
        Responce_heat("3", Integer.valueOf(seekBar_all.getProgress()));
    }
});

swich1.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
        int s1 = 0;
        if (swich1.isChecked()){
            s1 = 1;
        }else s1 = 0;

        int finalS1 = s1;
        Responce("pin4", finalS1);
    }
});
```

```
        swich2.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
                int s2 = 0;
                if (swich2.isChecked()){
                    s2 = 1;
                }else s2 = 0;
                int finalS2 = s2;
                Responce("pin5", finalS2);
            }
        });
        swich3.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton, boolean b) {
                int s3 = 0;
                if (swich3.isChecked()){
                    s3 = 1;
                }else s3 = 0;
                int finalS3 = s3;
                Responce("pin6", finalS3);
            }
        });
    }
    public void Responce(String pin, int status){
        new Thread(new Runnable() {
            @Override
            public void run() {

                RequestQueue requestQueue = Volley.newRequestQueue(getActivity().getApplicationContext());
                StringRequest stringRequest = new StringRequest(Request.Method.POST,
"https://bohdandiplom.000webhostapp.com/android_send.php", new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                    }
                }, new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast toast = Toast.makeText(getActivity().getApplicationContext(),
                                "Помилка запиту на сервер (ОПАЛЕННЯ вкл/викл): " + error.toString(),
Toast.LENGTH_LONG);
                        toast.show();
                    }
                }){
                    @Override
                    public Map<String, String> getParams() {
                        Map<String, String> map = new HashMap<>();
                        map.put("pin", pin );
                        map.put("bool", String.valueOf(status));
                        map.put("flag", "0");
                        return map;
                    }
                };
                requestQueue.add(stringRequest);
            }
        }).start();
    }
    public void Responce_heat(String heat, int target){
```

```
        new Thread(new Runnable() {
            @Override
            public void run() {

                RequestQueue requestQueue = Volley.newRequestQueue(getActivity().getApplicationContext());
                StringRequest stringRequest = new StringRequest(Request.Method.POST,
"https://bohdandiplom.000webhostapp.com/android_send_heat.php", new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                    }
                }, new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast toast = Toast.makeText(getActivity().getApplicationContext(),
                                "Помилка запиту на сервер (ОПАЛЕННЯ температура): " + error.toString(),
Toast.LENGTH_LONG);
                        toast.show();
                    }
                }){
                    @Override
                    public Map<String, String> getParams() {
                        Map<String, String> map = new HashMap<>();
                        map.put("heat", heat );
                        map.put("target", String.valueOf(target));
                        return map;
                    }
                };
                requestQueue.add(stringRequest);
            }
        }).start();
    }

    class updateTimer extends TimerTask{
        @Override
        public void run() {
            new Thread(new Runnable() {
                @Override
                public void run() {

                    RequestQueue requestQueue = Volley.newRequestQueue(getActivity().getApplicationContext());
                    StringRequest stringRequest = new StringRequest(Request.Method.POST,
"https://bohdandiplom.000webhostapp.com/android_update_heat.php", new Response.Listener<String>() {
                        @SuppressLint("SetTextI18n")
                        @Override
                        public void onResponse(String response) {
                            String ansver = response;
                            ansver = response;

                            if((ansver.charAt(0))%2==0){
                                swich1.setChecked(false);
                            } else swich1.setChecked(true);

                            if((ansver.charAt(2))%2==0){
                                swich2.setChecked(false);
                            } else swich2.setChecked(true);

                            if((ansver.charAt(4))%2==0){
```

```
                    swich3.setChecked(false);
                } else swich3.setChecked(true);
                temp1_1 = String.valueOf((ansver.charAt(6)));//Target temp, not current now (+2)
                temp1_2 = String.valueOf((ansver.charAt(7)));//Target temp, not current now (+2)
                temp2_1 = String.valueOf((ansver.charAt(10)));//Target temp, not current now (+2)
                temp2_2 = String.valueOf((ansver.charAt(11)));//Target temp, not current now (+2)
                temp3_1 = String.valueOf((ansver.charAt(14)));//Target temp, not current now (+2)
                temp3_2 = String.valueOf((ansver.charAt(15)));//Target temp, not current now (+2)
                swich1.setText( "Права частина будинку (" + temp1_1 + temp1_2 + "C" + ic_temp + ")");
                swich2.setText( "Ліва частина будинку (" + temp2_1 + temp2_2 + "C" + ic_temp + ")");
                swich3.setText( "Тепла підлога (" + temp3_1 + temp3_2 + "C" + ic_temp + ")");
                seekBar1.setProgress(Integer.parseInt(temp1_1+temp1_2));
                seekBar2.setProgress(Integer.parseInt(temp2_1 + temp2_2));
                seekBar3.setProgress(Integer.parseInt(temp3_1 + temp3_2));
                textView.setText(ansver);
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast toast = Toast.makeText(getActivity().getApplicationContext(),
                        getString(R.string.err_update_timer_heat) + error.toString(), Toast.LENGTH_LONG);
                toast.setGravity(Gravity.CENTER,0,0);
                toast.show();
            }
        }){
            @Override
            public Map<String, String> getParams() {
                Map<String, String> map = new HashMap<>();
                map.put("pin", "pin4" );
                map.put("pin2", "pin5" );
                map.put("pin3", "pin6" );
                map.put("heat", "1" );
                map.put("heat2", "2" );
                map.put("heat3", "3" );
                return map;
            }
        };
        requestQueue.add(stringRequest);
      }
    }).start();
  }
 }
}
```

ДОДАТОК Г

Відзив керівника економічного розділу

ДОДАТОК Д

ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ

Ім’я файла	Опис
Пояснювальні документи	
Диплом.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом.pdf	Пояснювальна записка до кваліфікаційної роботи. в форматі PDF
Програма	
Коди програм.RAR	Архів. Містить коди програм, скомпільовану програму, файли забезпечення серверу, програму для контролеру і дамп файл бази даних
Презентація	
Презентація.ppt	Презентація кваліфікаційної роботи.