

УДК 004.05

**К. Г. Онищенко¹, І. В. Афанасьєва², І. І. Арсеній³**^{1,2} ХНУРЕ, м. Харків, Україна, kostiantyn.onyshchenko@nure.ua³ Інститут загальної та невідкладної хірургії, м. Харків, Україна

РОЗРОБКА СЕРВІСУ ДЛЯ ЗАБЕЗПЕЧЕННЯ ДИСТАНЦІЙНОГО КЕРУВАННЯ ПРИЛАДАМИ ТА ПРИСТРОЯМИ РОЗУМНОГО ДОМУ «SMARTHOUSE» ІЗ ВИКОРИСТАННЯМ ІОТ РІШЕНЬ

Статтю присвячено проектуванню та реалізації сервісу керування пристроями розумного дому. Поетапно змодельована та спроектована архітектура програмної системи. Поетапно проаналізовано та узагальнено результати розробки, створено повноцінну робочу програмну систему. Відображені основні переваги спроектованої архітектури програмного забезпечення та прийнятих програмних рішень при розробці системи розумного дому з використанням ІоТ-рішень. Наведено прийняті архітектурні рішення та приклади роботи.

ІОТ (ІНТЕРНЕТ РЕЧЕЙ), РОЗУМНИЙ ДІМ, ДИСТАНЦІЙНЕ КЕРУВАННЯ ПРИЛАДАМИ

Вступ

Ми живемо у часи невинного розвитку та широкіх можливостей, коли щодня з'являються все нові й нові винаходи. Людина створює та вдосконалює їх задля покращення умов життя. Як наслідок, більшість сфер життя змушена переходити на новий рівень – рівень інформаційних технологій, щоб не зникнути як застаріла. Зараз майже все можна зникнути чи дізнатися за допомогою мережі Інтернет.

Дедалі більшої популярності набувають веб-системи. Такі системи мають певні переваги, а саме: доступність за допомогою мережі Internet, швидкість розповсюдження, менші вимоги до пам'яті комп'ютерів та іншої техніки, і т.д. Для даної роботи ці переваги мають ключове значення, тому була обрана саме веб-система.

З плином часу змінилися цінності та пріоритети людства: люди мандрують через весь світ, жінки працюють на рівних з чоловіками, і т.д. Тому дуже часто трапляється, що досить тривалий час дома ніхто не знаходиться, але за помешканням потрібен догляд. Це проявляється у елементарному: потрібно провітрювати приміщення, контролювати температурний режим, запускати прання, регулювати освітлення, тощо.

Вирішити ці питання покликані розумні прилади, що працюють за допомогою технології Internet of Things [1]. Це постійні інновації та велика кількість різноманітних проектів, які додають інтелекту і функціоналу різним пристроям. Досвід, накопичений розробниками ІоТ-індустрії за весь час її існування, дозволяє виділити загальні складові при створенні програмного продукту.

ІоТ-технологія здатна контролювати стан приладів та керувати ними за допомогою мережі інтернет [1]. Своєчасне повідомлення про стан приладів не тільки полегшує умови проживання, а й запобігає шкоді майну та мешканцям будинку. Усе разом це полегшує життя сучасної людини та робить його більш комфортним.

Отже, метою роботи є проектування, розробка сервісу для забезпечення дистанційного керування

приладами та пристроями розумного дому із використанням ІоТ рішень, рекомендацій та бібліотек ASP.NET MVC, Entity Framework та Java. Окремо слід спроектувати емулятор роботи розумних пристроїв [2]. Система має передбачати певну звітність результатів роботи розумних приладів, зручний і зрозумілий інтерфейс з нативними підказками для недосвідчених користувачів.

1. Постановка задачі

Необхідно спроектувати та реалізувати веб-сервіс для забезпечення дистанційного керування приладами та пристроями розумного дому із використанням ІоТ рішень.

Основне завдання проекту – вирішити проблему великої кількості сервісів для керування розумними приладами, що покривають різні сфери побуту.

Одна з головних ідей при розробці такої системи полягає у використанні структурованої методології при розробці сервісу розумного дому з використанням ІоТ-рішень. Така система повинна поєднувати у собі всі переваги та виключати недоліки вже існуючих систем за рахунок слідування формалізованим правилам, що чітко описують послідовність розробки програмного забезпечення з використанням ІоТ-рішень.

Створення єдиного веб-додатку повинно забезпечити надійне централізоване керування усіма розумними приладами, об'єднавши їх у систему розумного дому.

Інтерфейс програмної системи повинен мати сучасний дизайн, бути виконаним в одному кольоровому рішенні та мати інтуїтивно-зрозумілу систему навігації, керування та відстеження для користувача.

Емулятор приладу розумного дому повинен бути реалізований у консольному вигляді та відображати дані у режимі реального часу.

Основним джерелом доходів даного сервісу повинні бути сфери продажу та підтримки користувачів. Відсутність рекламного навантаження та постійна технологічна підтримка повинні привабити людей, що дорожать своїм часом.

2. Аналіз предметної галузі

Галузь інформаційних технологій сьогодення – керування розумними пристроями, використовуючи IoT-технологію. Вона, як найбільш ефективний спосіб обміну даними, здобуває все більшу і більшу популярність серед провідних виробників розумних пристроїв.

Інтернет речей (Internet of Things – IoT) – це концепція обчислювальної мережі фізичних предметів, оснащених вбудованими технологіями для взаємодії один з одним або з зовнішнім середовищем. Вона виключає з частини дій необхідність участі людини [2].

За своєю природою, розробка для IoT – це постійні інновації та велика кількість різноманітних проектів, які використовують штучний інтелект та функції різних пристроїв. Основною концепцією IoT є можливість підключення різних об'єктів (речей), які людина може використовувати в повсякденному житті, наприклад, холодильник, кондиціонер, автомобіль, велосипед і навіть кросівки. Всі ці об'єкти (речі) повинні бути оснащені вбудованими сенсорами, які мають можливість обробляти інформацію, що надходить з навколишнього середовища, обмінюватися нею і виконувати різні дії в залежності від отриманої інформації [3]. Прикладом впровадження такої концепції є система «розумний будинок» або «розумна ферма». Ця система аналізує дані навколишнього середовища і в залежності від показників регулює температуру в приміщенні. У зимовий період регулюється інтенсивність опалення, а в разі спекотної погоди будинок має механізми відкривання і закривання вікон, завдяки чому провітрюється будинок, і все це відбувається без втручання людини.

На сьогоднішній день існує велика кількість різних розумних гаджетів на кшталт розумних годинників, вагів, чайників, пилососів. Інформація про розумні прилади зберігається в основному у локальній базі даних, або на сервісі кожної конкретної організації [3]. Під час використання багатьох приладів одночасно виникають проблеми зі своєчасним відстеженням стану приладів та їх синхронізацією. Це створює певні незручності, що можуть стати поштовхом для відмови людини від користування розумними приладами.

Система керування розумного дому – це система, що дозволяє людям, не залежно від віку та статі, керувати усіма розумними пристроями, що знаходяться в його помешканні.

3. Моделювання та проектування програмної системи

Моделювання – це метод дослідження явищ і створення процесів, що ґрунтується на заміні конкретного об'єкта досліджень іншим, подібним до нього (моделлю) [3].

Для моделювання програмної системи використовувалась уніфікована мова UML, яка використовується у парадигмі об'єктно-орієнтованого

програмування та є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення [4]. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, названої UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем.

UML може бути застосовано на всіх етапах життєвого циклу аналізу систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, дають можливість представити систему у такому вигляді з якого її можна легко перевести в програмний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Саме тому для найбільш повного уявлення про програмну систему та її принципи дії, були побудовані UML діаграми, котрі дають повний опис системи [4].

Діаграма послідовності (англ. sequence diagram) відображає взаємодії об'єктів впорядкованих за часом. Зокрема, така діаграма відображає задіяні об'єкти та послідовність відправлених повідомлень між ними [4].

Діаграма послідовностей представляє собою систему в цілому та те, як відбувається взаємодія між об'єктами на різних рівнях [4]. Послідовність дій протягом усієї роботи з системою, починаючи від реєстрації нового користувача і закінчуючи зміною стану приладу розумного дому зображено на діаграмі послідовності, рис. 1.

На ній можна побачити, що відбувається в програмній системі. Коли користувач відкриває вікно з даними, відображення надсилає запит до контролера для отримання даних, контролер в свою чергу надсилає запит до моделі, яка отримує дані з бази даних. Ці дані з бази надходять до відображення та пройшовши обробку надаються користувачу. Коли він виконав зміни у стані приладу розумного дому і натиснув на кнопку підтвердження зміни стану, дані через відображення відправляються до контролера, який їх перевіряє та відправляє до моделі. Модель спочатку обробляє дані, а потім зберігає їх до бази. Контролер в свою чергу повертає користувачу сторінку з результатом, або перенаправляє до початкової сторінки. Це загальна схема послідовності роботи будь-якого користувача з системою.

Тобто коли користувач просто відриває сторінку з даними, то контролер запрошує дані у моделі, яка отримує їх з бази та повертає, контролер обробляє дані та повертає дані для відображення.

Після перевірки наданої інформації дані заносяться у базу даних для подальшого використання.

Архітектура програмного забезпечення (англ. software architecture) – спосіб структурування програмної або обчислювальної системи, абстракція елементів системи на певній фазі її роботи. Система може складатись з кількох рівнів абстракції, і мати багато фаз роботи, кожна з яких може мати окрему архітектуру. Правильне проектування архітектури

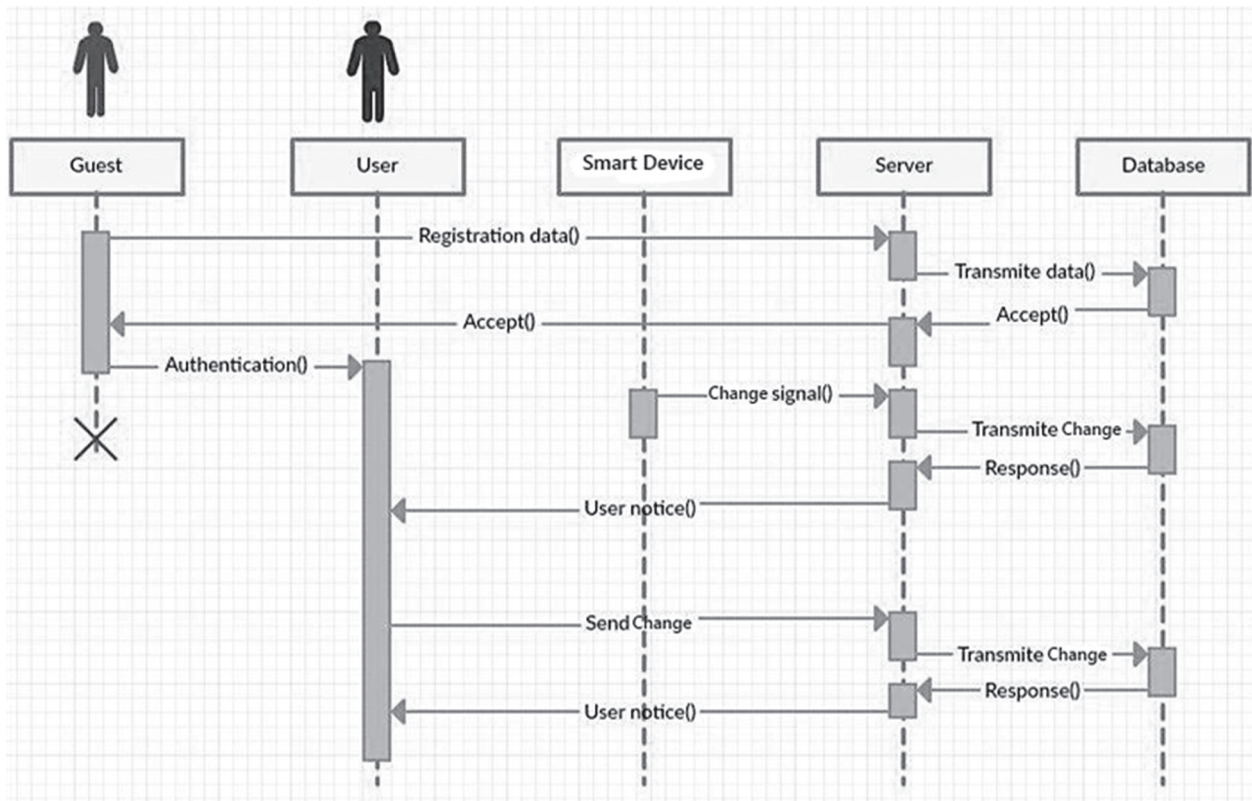


Рис. 1. Діаграма послідовності

програмної системи робить систему зданю до розширення та полегшує її супровід під час подальшого використання.

З огляду на всі переваги і недоліки вже існуючих рішень необхідно використовувати всі переваги об'єктно-орієнтованого підходу в програмуванні для створення гнучкої архітектури програмної системи. Це значно підвищить витрати на ранніх етапах розробки продукту, проте дозволить мінімізувати їх на стадії супроводу готового рішення. Досягнення гнучкості в архітектурі досягається побудовою концептуально правильної діаграми класів використовуючи основні парадигми ООП і SOLID принципи, рис. 2.

Абстрактні класи та інтерфейси дозволяють користувачеві додавати нові елементи розумного будинку шляхом виділення загальних методик і властивостей у вже існуючих (закладених) наборів поведінки, рис. 3.

4. Опис прийнятих програмних рішень

Для реалізації веб-сервісу було вирішено використовувати клієнт-серверну архітектуру. Цей архітектурний підхід є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережових додатків і передбачає взаємодію та обмін даними між ними [4].

У такій системі сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє

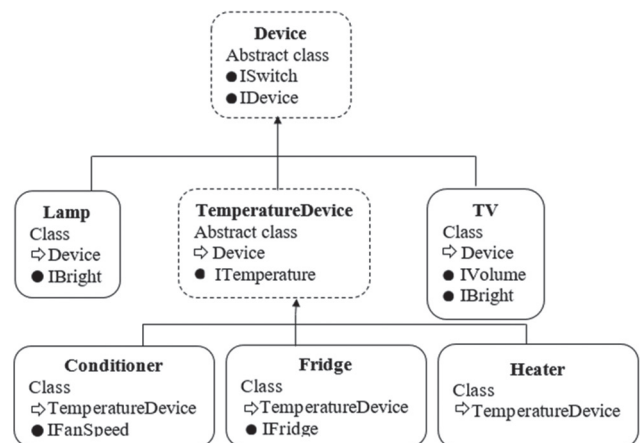


Рис. 2. Діаграма класів

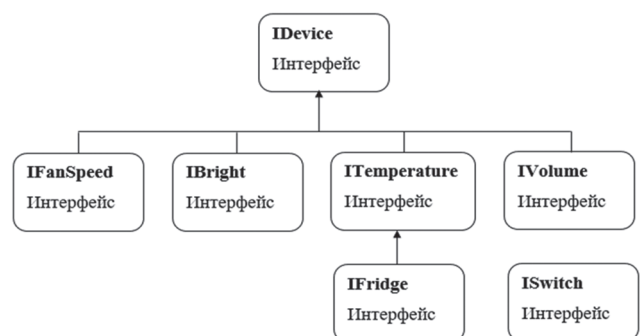


Рис. 3. Реалізація інтерфейсів

запити від різних клієнтів, з другого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Для веб-серверу, який реалізує сервіс було обрано технологію ASP.NET MVC. Цю технологію було обрано, тому що вона являється однією з найбільш функціональних та зручних для розробки веб-програм, актуальною і популярною серед розробників. Технологія ASP.NET MVC базується на шаблоні проектування MVC.

Паттерн архітектури Model-View-Controller (MVC) поділяє програму на три основні частини: модель, уявлення і контролер. Розділення це здійснюється як на фізичному так і на логічному рівнях.

Використання MVC шаблону надає можливість будувати впорядковану програмну систему та дозволяє попередити та запобігти плутаниці в програмному коді, котра може виникнути в проектах значного масштабу.

Отриманий інтерфейс програмного продукту є інтерактивним та максимально інформативним. Він відображає користувачу інформацію про наявні прилади та їх стан, рис. 4.

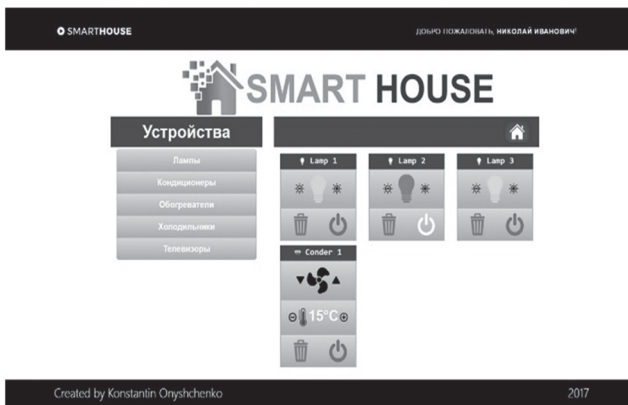


Рис. 4. Головна сторінка користувача

Для взаємодії з БД користувачів було обрано технологію Entity Framework, яка дозволяє розробникам розробляти додатки для доступу до даних, що працюють з концептуальною моделлю програми, а не безпосередньо з реляційною схемою зберігання даних.

Для розробки БД за допомогою Entity Framework було обрано стратегію Code First («спочатку код»). Підхід Code First дозволяє визначити модель з використанням C# класів, яка буде використана для створення схеми бази даних або для зіставлення з існуючою базою даних. Таким чином, використовуючи запити до БД користувачеві надається звіт про роботу приладу розумного дому, рис. 5.

Для зберігання інформації використовується база даних Microsoft SQL Server.

Розроблений емулятор створювався за концепцією IoT (Інтернет речей), рис. 6. Вона полягає у оснащенні фізичних об'єктів (речей) вбудованими технологіями для взаємодії один з одним або зовнішнім середовищем [4]. Організація таких мереж має на меті перебудувати економічні та суспільні процеси, зменшуючи необхідність безпосередньої участі людини.



Рис. 5. Перегляд результатів роботи конкретного пристрою розумного дому

У якості мови програмування була обрана Java через те що програми, написані на ній, транслюється в байт-код, що виконується віртуальною машиною, оброблює її та передає інструкцію обладнанню як інтерпретатор [4]. Перевагою подібного способу виконання програм є повна незалежність байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки, в рамках якої виконання програми повністю контролюється віртуальною машиною.

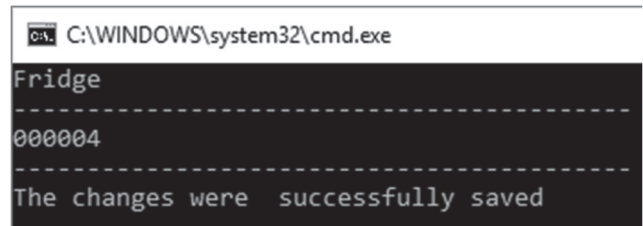


Рис. 6. Робота емулятору пристрою розумного дому

Оскільки, як ідентифікаторів IoT-пристроїв використовуються MAC-адреса мережевого адаптера, особливу увагу було приділено коректній ідентифікації по протоколу IPv6, який забезпечує унікальні адреси мережевого рівня не менше 300 млн. пристроїв на одного мешканця планети.

Для бездротової передачі даних особливо важлива роль в побудові інтернету речей грають такі якості, як ефективність в умовах низьких швидкостей, відмовостійкість, адаптивність, можливість самоорганізація. За досягнення даних цілей відповідає стандарт IEEE 802.15.4 [5].

Серед провідних технологій важливу роль в інтеграції інтернету речей грають рішення PLC – технології побудови мереж передачі даних по лініях електропередач, так як у багатьох додатках присутній доступ до електромереж (наприклад, торгові автомати, банкомати, інтелектуальні лічильники, контролери освітлення спочатку підключені до мережі електропостачання), 6LoWPAN, який реалізує шар IPv6 як над IEEE 802.15.4, так і над PLC, будучи відкритим протоколом, по стандарту IETF, відзначається як особливо важлива для розвитку інтернету речей

Висновки

Створений сервіс для забезпечення дистанційного керування приладами та пристроями розумного дому дозволяє контролювати стан розумних приладів та впливати на їх роботу шляхом зміни налаштувань у веб додатку в незалежності від місця розташування користувача. Це, без сумнівів, може значно підвищити рівень комфорту проживання у помешканні та зменшити значну кількість часу.

Розроблена система є корисним сервісом для людей, що цінують свій час, націлені на покращення своїх житлових умов та мають жагу до впровадження новітніх технологій у своє повсякденне життя. Без сумнівів, система розумного дому не тільки економить час користувача, а ще й підвищує рівень безпеки його помешкання. Сервіс дозволяє слідкувати за станом датчиків та приладів розумного дому дистанційно, де б не знаходився користувач.

Список літератури:

1. IOT Documentation [Електронний ресурс] / Intel Software Developer Zone. – Режим доступу: <https://software.intel.com/en-us/articles/iot-path-to-product-how-to-build-the-smart-home-prototype/> 28.04.2017 р. – Загол. з екрану.
2. *Velusami K.* Iinternet of Things in Cloud IDG/ K. Velusamy, D. Venkitaramanan, Sh. K. Vasudevan// *Journal of Engineering and Applied Sciences.* – 2013. – 8(9-12). – P. 304-313.
3. *Онищенко К. Г.* Методология разработки систем умного дома с применением IoT решений / К. Г. Онищенко, И. В. Афанасьєва // *Радиоэлектроника и молодежь в XXI веке: материалы 21-го Междунар. молодежн. форума.* – г. Харьков: ХНУРЭ, 2017. – С. 157-158.
4. *Фаулер М.* UML Основы / М. Фаулер, К. Скотт – СПб: Символ-Плюс, 2002. – 192 с.
5. *Онищенко К. Г.* Структурированная методология разработки сервиса для обеспечения дистанционного управления приборами и устройствами умного дома с использованием Internet of Things решений / К.Г.Онищенко, И. В. Афанасьєва // *ScienceRise.* – 2017. – 5(2). – С. 30-33.

Надійшла до редколегії 23.06.2017