

СТРУКТУРА МОБІЛЬНОГО ІНТЕРФЕЙСУ СИСТЕМИ КОНТРОЛЮ СТАНУ РОЗУМНОГО БУДИНКУ

Вінницький національний технічний університет.

Анотація

Запропоновано варіант розробки мобільного інтерфейсу системи з із мобільним адаптивним інтерфейсом, що дозволяє, здійснювати контроль за параметрами розумного будинку через мережу інтернет із різних пристроїв.

Ключові слова: розумний будинок, інтерфейс, людина, мобільний пристрій

Abstract

Having been proposed method development of a mobile system interface with a mobile adaptive interface, which allows monitoring the parameters of a smart home via the Internet from various devices.

Keywords: smart home, interface, human, mobile device.

Вступ

Інформаційні технології за останні десятиліття зробили великий крок вперед у своєму розвитку і вплили, практично, у всі сфери життя людини, не залишивши осторонь і власну оселю. Наявність таких засобів автоматизації як давачі, автоматичні вимикачі, мікроконтролери, різні інформаційно-вимірювальні прилади, сенсори, комп'ютерні мережі і тому подібних у поєднанні з сенсорними панелями, персональними комп'ютерами або ж із мобільними пристроями надають можливість створювати різного рівня складності системи контролю за станом будинку. Розумний будинок – система пристроїв, що розташовані у будинку, які здатні виконувати задачі контролю за станом будинку без участі людини. Поки що такі системи досить кошторисні тому їх можуть лише організації або заможні особи. Але в недалекому майбутньому, як свідчить історія розвитку комп'ютерних технологій, ставатимуть більш дешевими та доступними для ширшого кола осіб.

Вже зараз підприємства починають пропонувати свої послуги по впровадженню розумних будинків для захоплення своєї ніші на цьому ринку. Але все ще таких гравців не так багато, і увійти на цей ринок із своєю особливою пропозицією по впровадженню системи розумного будинку.

Окремої уваги вимагають інтерфейси керування такою системою, оскільки від них напряму залежить комфорт керування. Тому вирішено дослідити проблематику розробки інтерфейсів та сучасні методи розробки.

Результати дослідження

Визначимо основні функції інтерфейсу:

- авторизація користувача;
- реалізацій керування користувачами через адмін-панель;
- розподілення користувачів по дозволеному функціоналу;
- перегляд поточних значень контролюючих параметрів;
- перемикання функцій розумного будинку.
- Веб-інтерфейс, що розробляється повинен відповідати вимогам:
- адаптивність;
- комфорт та простота використання.

Для розробки сучасного гнучкого інтерфейсу потрібен фреймворк, що надасть відповідний інструментарій розробки.

Bootstrap – надає вільний вибір інструментів для створення сайтів та веб-додатків. Включає в себе HTML та CSS шаблони оформлення для типографіки, веб-форм, кнопок, блоків навігації та інших

компонентів веб-інтерфейсу, включаючи JavaScript-розширення. Варто зазначити, що фреймворк може погано підтримуватися старими браузерами[1]

Враховуючи велику підтримку Bootstrap з боку спільноти, що в свою чергу означає більшу кількість документації по роботі фреймворку, то оберемо саме його.

Для зберігання даних, які відобразитиме інтерфейс потрібна база даних. У якості такої можуть виступати Microsoft SQL Server, MySQL, Oracle, PostgreSQL. Оскільки MySQL підтримується більшістю мов програмування, то обираємо саме її.

Щоб надати функціональності розроблюваному інтерфейсу та надати доступ йому до бази даних потрібно використати одну із наявних мов писання сценаріїв. Найбільш популярними на даний момент є дві мови – PHP та ASP.NET.

Порівняльна характеристика мов представлена у таблиці 1

Таблиця 1 – порівняльна характеристика мов.

Хар-ка	ASP.NET	PHP
Простота	Простота розробки як простих так і складних проєктів.	При розробці складної системи ця перевага перетворюється на недолік
Коштовність	Якщо для сайту достатньо одного сервера, то вартістю ОС можна знехтувати.	Повна безкоштовність, включаючи платформу.
Швидкодія	Код компілюють, завдяки чому він виконується набагато швидше	Відсутня компіляція, відповідно швидкодія менша.
Надійність	Більш надійний при обробці та відображенні даних	Суттєво менша (проблеми кодування, неповноцінність мови).

Як видно, суттєвих переваг одна мова над іншою немає, але взявши до уваги ту особливість, що PHP має більшу підтримку хостингів та кращу документованість, то оберемо саме цю мову.

Розробимо базу даних, що дозволить нам зберігати наші дані.

Таблиця 2 – представлено перелік всіх полів таблиці даних.

	Ім'я поля	Тип
1.	Ім'я користувача(username)	Varchar(255)
2.	Пароль користувача(userpassword)	Varchar(255)
3.	Назва функціоналу(funcName)	Varchar(255)
4.	Код піктограми функціоналу(divFunc)	Text
5.	Назва кімнати(nameRoom)	Varchar(255)
6.	Температура у кімнаті(temperatureValue)	Int(11)
7.	Стан функціоналу кімнати(isActivated)	Tinyint(1)
8.	Контрольований параметр(nameParameter)	Varchar(255)
9.	Значення контрольованого параметру(Value)	Int(11)

Таблиця 3 – Приклад можливого зберігання даних без нормалізації

Ім'я поля	Запис 1	Запис 2	Запис 3
username	Son	Douther	mother
userpassword	123	321	231
funcName	Light	Light	Alarm clock
divFunc	1	1	5
nameRoom	Bedroom	Outside	Bedroom
temperatureValue	25	26	27
isActivated	1	0	1
nameParameter	Gas, electricity	Gas	Electricity

ValueParameter	2, 3	3	5
----------------	------	---	---

Як бачимо із таблиці 3 багато даних повторюються. Така ситуація приведе до витрати значного часу на введення повторювальних даних, необмеженому збільшенню бази даних, збільшиться кількість помилок при введенні.

Для задовільнення умов першої нормальної форми таблиця повинна мати принаймні один унікальний ключ. Тому для виконання умови необхідно визначити ключ, котрий включав би сукупність всіх полів таблиці, однозначно визначав запис. Повинен усувати повторювані групи полів[2]

Розіб'ємо загальну таблицю на таблиці Users та FunctionalOfRooms. На рисунку представлено 1 показано розділення. Між таблицями буде зв'язок один до багатьох по коду користувача.

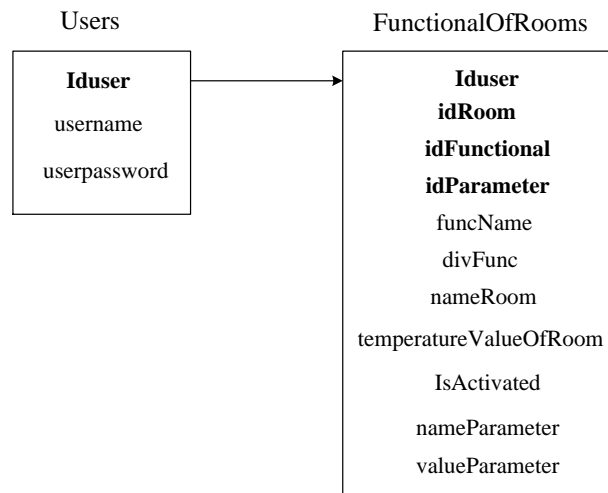


Рис. 1. Таблиці БД у першій нормальної формі

Поняття 2-ї нормальної форми може застосовуватися до таблиць, що мають складений ключ. У таблиці FunctionalOfRooms є складений ключ: idUser, idRoom, idFunctional, idParameter. Для приведення до 2-ї нормальної форми створимо таблиці Rooms, Functional, ControlParameters. На рисунку 2 представлено таблиці у 2-й нормальної формі.

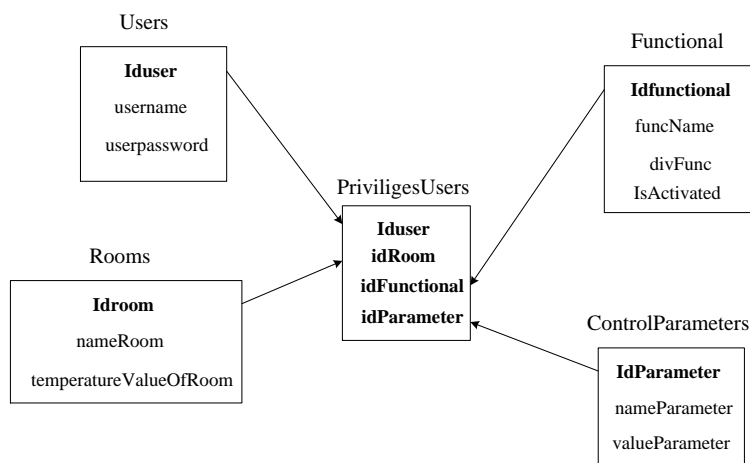


Рис. 2. Таблиці у другій нормальної формі

Оскільки значень контрольованих параметрів може бути багато, то варто винести значення параметрів в окрему таблицю. Після цього, можна сказати, що всі поля таблиць однозначно визначені. На рисунку 3 зображено таблиці у третій нормальної формі.

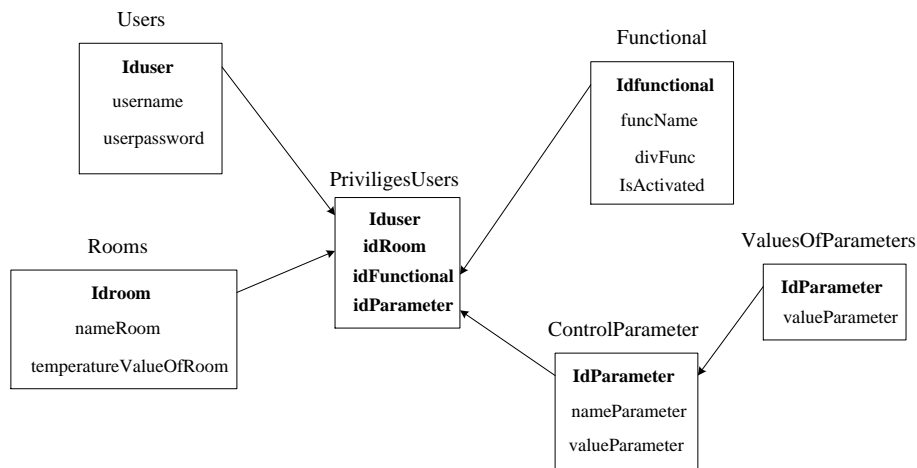


Рис. 3. Таблиці у третій нормальній формі.

Наступний етап розробки бази даних – оптимізація для більш зручного зберігання даних. Відповідно до логіки програми, кожен користувач має мати доступ для певного функціоналу у кімнаті, а також його стану. Тому доцільно створити ще одну таблицю FunctionalOfRooms, яка міститиме в собі ключ на запис таблиці Functional and Rooms та поле isActivated для визначення стану функціоналу. А таблиця PriviligesUsers в свою чергу зберігатиме ключ на запис таблиці Users та FunctionalOfRooms.

Побудуємо ER-модель бази даних. Для виділимо такі сутності: Користувачі, Привілегії користувачів, Функціонал, Кімнати, Функціонал кімнат, Контрольовані параметри, Значення контрольованих параметрів

Наступний етап – це розробка алгоритму функціонування системи.

Для користувача системи визначаються наступні функції: авторизація, перегляд значень контролюючих параметрів, зміна стану об'єктів кімнати.

Адміністратор системи має тіж функції, що і користувач плюс додатково може створювати нових користувачів, видаляти їх, призначати їм привілегії.

На рисунку 4, 5, 6, 7, 8, 9, 10 показано діаграму діяльностей процесу авторизації, отримання стану об'єктів, зміни стану об'єктів, статистики, створення користувача, видалення користувача та призначення йому прав відповідно.

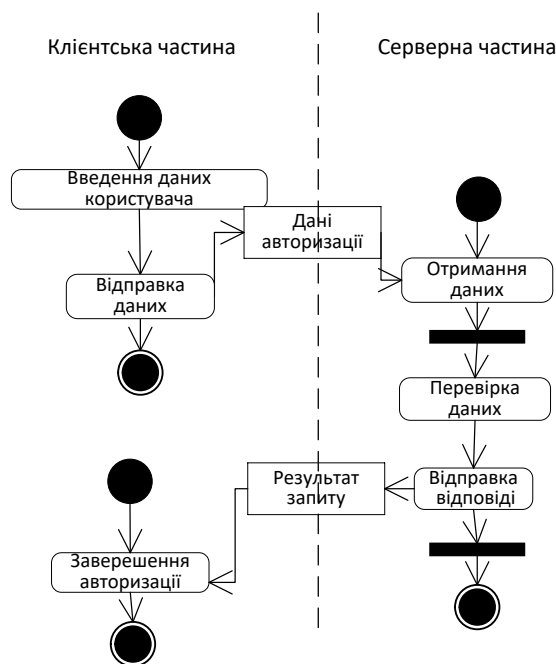


Рис. 4. Діаграма діяльності процесу авторизації

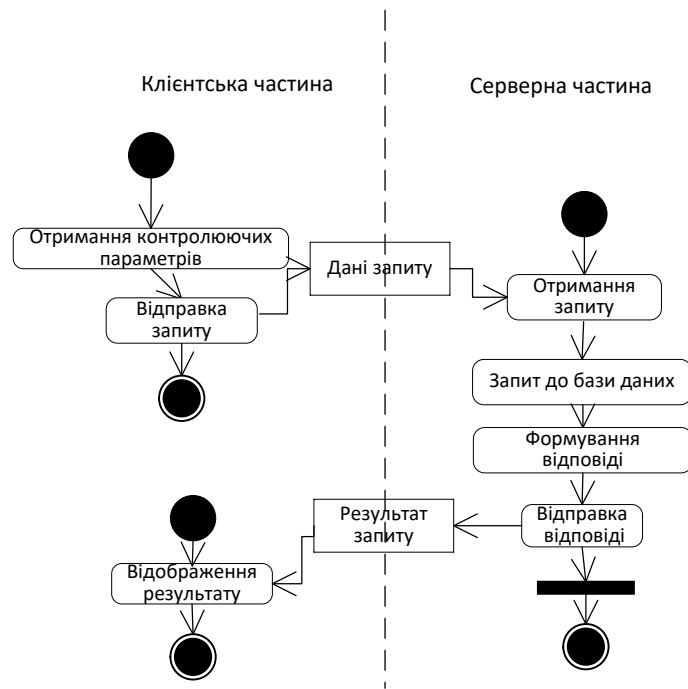


Рис. 5. Діаграма діяльності процесу отримання стану об'єкту

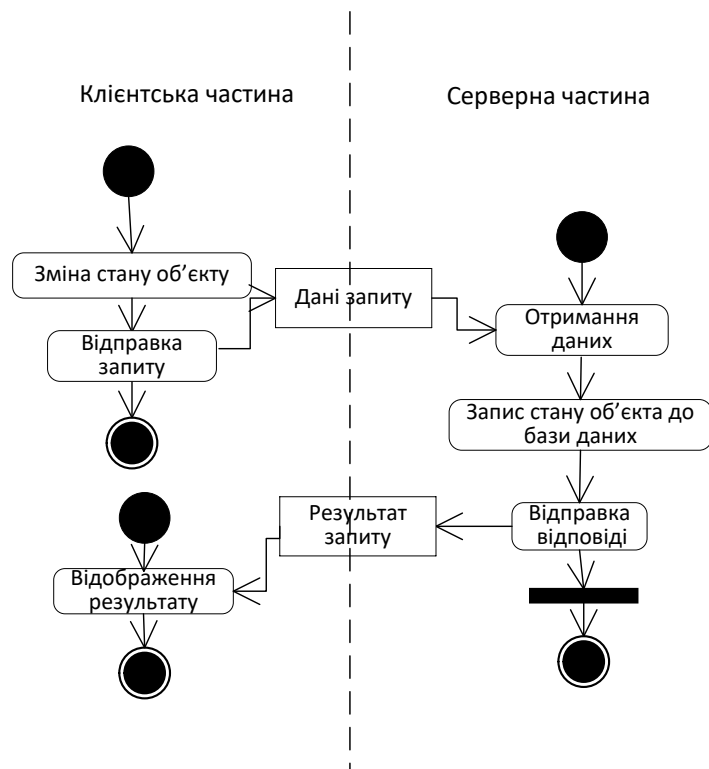


Рис. 6. Діаграма діяльності процесу зміни стану об'єкту

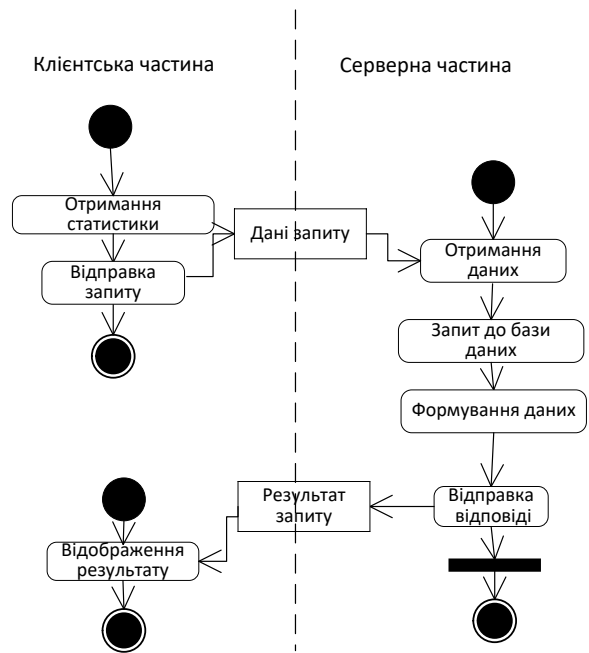


Рис. 7. Діаграма діяльності процесу отримання статистики

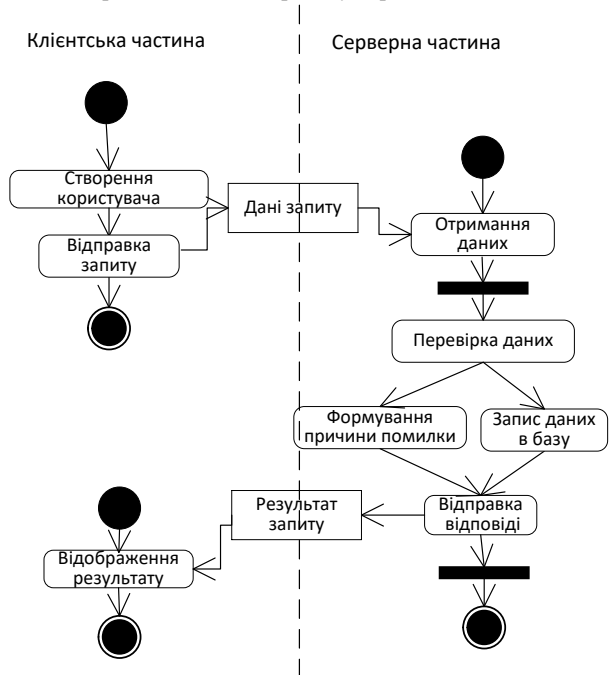


Рис. 8. Діаграма діяльності процесу створення нового користувача

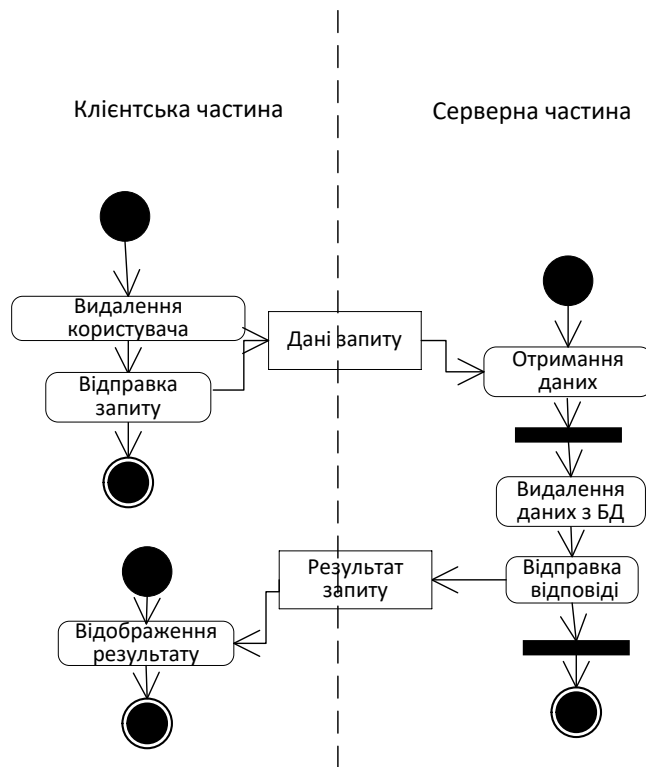


Рис. 9. Діаграма діяльності процесу видалення користувача

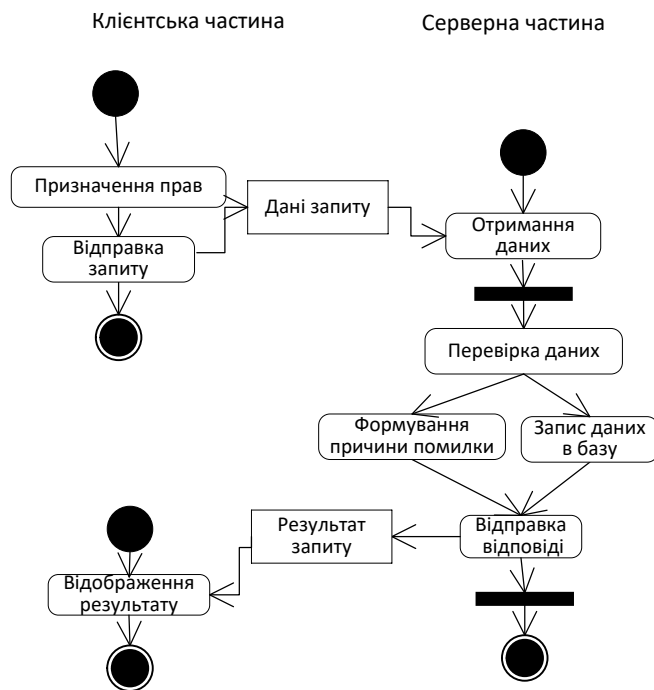


Рис. 10. Діаграма діяльності призначення прав користувачеві

Висновки

В результаті проведеного дослідження вирішено, що для розробки даної системи буде використано серверну мову PHP, система управління базами даних MySQL, фреймворк Bootstrap. Також у системі буде реалізовано детальну статистику по контролюючим параметрам.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Bootstrap(фреймворк) - Вікіпедія[Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : Creative Common Attribution-ShareAlike. – Режим доступу: [https://ru.wikipedia.org/wiki/Bootstrap_\(фреймворк\)](https://ru.wikipedia.org/wiki/Bootstrap_(фреймворк)) – Назва з екрана.

2. Нормалізація таблиць реляційної бази даних - Реферати, курсові, дисертації, дипломи [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – : 2018. – Режим доступу: http://ua-referat.com/Нормалізація_таблиць_в_реляційної_моделі_бази_даних – Назва з екрана.

Олександр Геннадійович Желюк – студент групи 2АКІТ-18м, факультет комп'ютерних систем та автоматики, м. Вінниця, e-mail: mhammer1041@gmail.com;

Науковий керівник: **Володимир Михайлович Дубовой** – доктор технічних наук, професор, завідувач кафедри комп'ютерних систем управління, Вінницький національний технічний університет, м. Вінниця

Oleksandr H. Zheliuk – departure of Computer Systems and Automation, Vinnytsia National Technical University, Vinnytsia, email: mhammer1041@gmail.com;

Supervisor: **Vladimir M. Dubovoy** – Dr. Sc(eng.), Professor, Head of chair of Computer control systems, Vinnytsia National Technical University, Vinnytsia.