НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ПОГОДЖЕНО **Декан факультету**	**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ** **Завідувач кафедри інформаційних систем і технологій**
_____ Ігор БОЛБОТ (підпис)	**_____ Михайло ШВИДЕНКО** (підпис)
«___» _____ 2026 р.	«___» _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему: «ІНФОРМАЦІЙНА СИСТЕМА КОНТРОЛЮ ВІДВІДУВАЧІВ ДЛЯ SMART CAMPUS НА ОСНОВІ HOME ASSISTANT»

Спеціальність «122 Комп'ютерні науки»
Освітньо-професійна програма «Комп'ютерні науки»

Гарант освітньої програми
д.е.н., професор **_____ Роман РУДЕНСЬКИЙ**

Керівник бакалаврської кваліфікаційної роботи **_____ Тетяна ВОЛОШИНА**
к.п.н., доцент

Виконав **_____ Денис МЕДЯНИК**

Київ-2026

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри інформаційних систем і технологій

к.е.н., доцент _____ Михайло ШВИДЕНКО

«___» _____ 2026 р.

ЗАВДАННЯ ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ

Медяник Денису Андрійовичу

Спеціальність «122 Комп'ютерні науки»
Освітньо-професійна програма «Комп'ютерні науки»
Тема бакалаврської кваліфікаційної роботи
«**ІНФОРМАЦІЙНА СИСТЕМА КОНТРОЛЮ ВІДВІДУВАЧІВ ДЛЯ SMART CAMPUS НА ОСНОВІ HOME ASSISTANT**»
затверджена наказом від «26» січня 2026 р. № 46 «С»
Термін подання завершеної роботи на кафедру «22» травня 2026 р.
Вихідні дані до бакалаврської кваліфікаційної роботи: Методичні рекомендації кафедри, наукові та практичні розробки працівників кафедри, довідкові матеріали по системі Home Assistant
Перелік питань, що підлягають дослідженню:
Проаналізувати предметну область та існуючі підходи до організації контролю доступу в середовищі Smart Campus; дослідити можливості платформи Home Assistant для реалізації підсистеми контролю відвідувачів; сформувати функціональні вимоги до системи; розробити UML-моделі функціонування системи; спроєктувати логічну та фізичну моделі бази даних.
Дата видачі завдання «28» січня 2026 р.

Керівник бакалаврської кваліфікаційної роботи _____ **Тетяна ВОЛОШИНА**

Завдання взяв до виконання _____ **Денис МЕДЯНИК**

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API − Application Programming Interface, програмний інтерфейс застосунку

DB − Database, база даних

ESP32 − мікроконтролерна платформа сімейства Espressif

ER-модель − модель «сутність-зв’язок»

NFC − Near Field Communication, технологія безконтактного обміну даними на близькій відстані

PK − Primary Key, первинний ключ

RFID − Radio Frequency Identification, радіочастотна ідентифікація

SQL − Structured Query Language, мова структурованих запитів

UML − Unified Modeling Language, уніфікована мова моделювання

UID − Unique Identifier, унікальний ідентифікатор мітки

YAML − формат представлення конфігураційних даних

ВСТУП

У сучасних умовах цифрової трансформації закладів вищої освіти все більшої актуальності набуває концепція Smart Campus, що передбачає інтеграцію інформаційних технологій у навчальне, адміністративне та безпекове середовище університету. Одним із важливих напрямів розвитку такої концепції є автоматизація контролю доступу до приміщень, облік подій проходу та централізоване керування пов’язаними процесами.

Актуальність теми полягає в необхідності створення інформаційної системи контролю відвідувачів, яка могла б бути інтегрована в інфраструктуру Smart Campus, забезпечувати ідентифікацію користувачів, перевірку прав доступу, реєстрацію подій проходу, формування звітної інформації та підтримку подальшого розширення функціональних можливостей.

Як основу реалізації у роботі обрано платформу Home Assistant, яка надає можливість інтегрувати апаратні пристрої, керувати автоматизаціями, відображати події та стани в єдиному інтерфейсі, а також поєднувати серверну логіку, інтерфейс користувача й моніторинг системи в одному середовищі. Для практичної реалізації підсистеми використовується мікроконтролерний вузол на базі ESP32 та інтеграція через ESPHome. У демонстраційному макеті застосовується зчитувач RC522, однак для реального впровадження більш доцільним є використання PN532.

Метою бакалаврської роботи є розробка інформаційної системи контролю відвідувачів для Smart Campus на основі Home Assistant, яка забезпечує ідентифікацію користувачів за RFID/NFC-мітками, перевірку прав доступу, журналювання подій проходу, відображення інформації в інтерфейсі користувача та підтримку подальшого розширення функціональності системи.

Для досягнення поставленої мети необхідно вирішити такі завдання: проаналізувати предметну область та існуючі підходи до організації контролю доступу в середовищі Smart Campus; дослідити можливості платформи Home

Assistant для реалізації підсистеми контролю відвідувачів; сформувати функціональні вимоги до системи; розробити UML-моделі функціонування системи; спроєктувати логічну та фізичну моделі бази даних; обґрунтувати вибір PostgreSQL як системи управління базою даних; реалізувати програмно-апаратну структуру системи на базі Home Assistant, ESPHome та ESP32; реалізувати інтерфейс користувача, журнал подій та механізми формування звітно-статистичної інформації; проаналізувати результати роботи системи та визначити напрями її подальшого вдосконалення.

Об’єктом дослідження є процес організації контролю відвідувачів у цифровому середовищі Smart Campus. Предметом дослідження є методи, моделі, програмні та апаратні засоби побудови інформаційної системи контролю відвідувачів на основі Home Assistant.

Практичне значення одержаних результатів полягає в розробці структури та прототипу підсистеми контролю відвідувачів, яка може бути використана як складова Smart Campus.

Додатковою підставою для актуальності теми є зміна очікувань користувачів університетського середовища. Студенти й працівники звикли до цифрових сервісів, що працюють швидко, передбачувано й без зайвих паперових процедур. Саме тому система контролю відвідувачів повинна не лише обмежувати доступ, а й бути частиною загальної логіки цифрового кампусу, де всі дії мають зрозумілий інформаційний супровід.

Важливо враховувати і те, що в умовах сучасного університету проходження до приміщень не є суто технічним питанням. Воно пов’язане з організацією потоків людей, безпекою обладнання, контролем відвідування окремих зон, а також можливістю швидко відновити послідовність подій у випадку нестандартної ситуації. Саме тому підсистема контролю відвідувачів має бути побудована як інформаційне рішення, а не лише як електронний замок із локальним зчитувачем.

Ще одним чинником є доцільність використання відкритих програмних платформ і доступних апаратних компонентів у навчально-науковому середовищі. Це дає можливість не лише створити практичний прототип, а й у подальшому адаптувати його до реальних умов експлуатації без жорсткої залежності від дорогих пропрієтарних комплексів.

Наукова новизна отриманих результатів у межах бакалаврської роботи полягає у запропонованому підході до побудови підсистеми контролю відвідувачів не як окремого автономного рішення, а як інтегрованого модуля вже існуючого цифрового середовища Smart Campus. На відміну від типових локальних систем доступу, запропонована модель орієнтована на використання єдиної платформи Home Assistant, яка поєднує серверну обробку, інтерфейс користувача, журналювання та інтеграцію з апаратними вузлами.

Практична цінність роботи визначається тим, що розроблена структура може бути використана не лише для демонстрації окремих сценаріїв доступу, а і як база для подальшого етапного впровадження в умовах навчального закладу. Це означає, що результати роботи можуть бути корисними не лише в навчальному процесі, а й під час побудови пілотного контуру системи контролю доступу в реальному університетському середовищі.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПІДХОДІВ ДО ОРГАНІЗАЦІЇ КОНТРОЛЮ ДОСТУПУ В SMART CAMPUS

1.1. Поняття та особливості концепції Smart Campus

У сучасних умовах цифровізації освітнього простору все більшої ваги набуває концепція Smart Campus, яка передбачає створення інтегрованого інформаційно-технологічного середовища закладу освіти. На відміну від традиційного підходу, де окремі цифрові сервіси існують відокремлено, Smart Campus орієнтується на взаємодію підсистем у межах єдиного керованого простору.

У загальному вигляді Smart Campus можна визначити як сукупність програмних платформ, мережевої інфраструктури, сенсорів, виконавчих пристроїв, сервісів моніторингу та аналітичних засобів, що забезпечують підтримку навчальної, адміністративної й безпекової діяльності університету. Основна ідея полягає не лише в автоматизації окремих дій, а й у накопиченні даних та їх використанні для прийняття управлінських рішень.

До типових складових Smart Campus належать системи відеоспостереження, контролю доступу, енергоменеджменту, обліку ресурсів, інформаційного оповіщення, моніторингу параметрів середовища та централізовані панелі адміністрування. У такому середовищі підсистема контролю відвідувачів займає важливе місце, оскільки безпосередньо пов’язує фізичний простір навчального закладу з його цифровою інфраструктурою.

Важливою рисою Smart Campus є масштабованість. Якщо на початковому етапі система охоплює лише окремий корпус або одну точку проходу, її архітектура все одно повинна допускати подальше розширення без повного перепроєктування.

Саме тому підсистему контролю відвідувачів доцільно відразу розробляти як модуль ширшого інформаційного середовища.

Підсистема контролю відвідувачів є однією з ключових складових безпекової інфраструктури сучасного закладу освіти. Її призначення полягає не лише у фізичному дозволі чи блокуванні проходу, а й у забезпеченні впорядкованого режиму доступу, розмежуванні прав користувачів, фіксації подій та створенні інформаційної основи для подальшого аналізу.

Університетське середовище має кілька типових категорій користувачів: студенти, співробітники, адміністрація, охорона, відвідувачі та гості. Для кожної з цих категорій правила доступу можуть відрізнятися за зонами, часовими межами та рівнем привілеїв. Саме тому система має бути достатньо гнучкою, щоб працювати з ролями, зонами, статусами міток і часовими інтервалами.

У межах Smart Campus система контролю доступу має розглядатися як інформаційна система. Це означає, що вона повинна ідентифікувати користувача, перевіряти його права, фіксувати історію проходів, формувати інцидентні повідомлення та підтримувати централізоване адміністрування. Лише за таких умов контроль доступу перестає бути ізольованим механізмом і стає частиною єдиної цифрової екосистеми університету.

1.2. Аналіз існуючих технологій ідентифікації та контролю доступу

Для організації контролю доступу використовуються різні технології ідентифікації: RFID, NFC, PIN-коди, QR-коди та біометричні методи. Кожен із цих підходів має власні переваги та обмеження, тому вибір конкретного рішення залежить від вимог до безпеки, вартості впровадження, зручності користування та можливості інтеграції в існуючу інфраструктуру.

RFID як поширений засіб ідентифікації

RFID є одним із найбільш доступних і технічно простих способів ідентифікації в системах доступу. Технологія базується на використанні безконтактної мітки, яка містить унікальний ідентифікатор і зчитується спеціальним модулем.

До переваг RFID належать невисока вартість, простота побудови прототипу, достатня швидкість зчитування та доступність модулів і міток. До недоліків варто віднести обмеженість сучасних сценаріїв взаємодії та меншу гнучкість у порівнянні з NFC.

NFC як перспективний напрям розвитку системи

NFC можна розглядати як сучасніший і перспективніший напрям порівняно з базовим RFID. Ключовою перевагою NFC є підтримка ширшого кола сценаріїв, у тому числі потенційна взаємодія зі смартфонами та іншими мобільними пристроями.

Саме тому в роботі рекомендовано використовувати PN532 як перспективний варіант практичного впровадження системи.

PIN-коди та паролі

PIN-коди є простим способом контролю доступу, який не потребує окремих міток. Однак їхнім суттєвим недоліком є можливість передачі коду іншій особі або його компрометація. Для Smart Campus такий підхід не є оптимальним як основний механізм доступу.

QR-коди

QR-коди можуть бути корисними для тимчасових сценаріїв: гостьового доступу, разових перепусток або вхідних квитків на події. Проте як постійний засіб доступу для студентів чи працівників вони менш практичні, ніж RFID або NFC.

Біометричні методи

Біометричні методи забезпечують високий рівень персоналізації, однак потребують складнішого обладнання, більших витрат і додаткової уваги до захисту персональних даних. Для даного проєкту вони не є доцільними як базове рішення.

Порівняльна оцінка підходів

Для системи контролю відвідувачів у Smart Campus найбільш раціональним є RFID/NFC-підхід, оскільки він достатньо простий для прототипування, зручний у повсякденному використанні та добре інтегрується з мікроконтролерами.

Отже, у демонстраційному макеті доцільно використовувати RC522, а для практичного впровадження − PN532 як більш перспективний засіб.

1.3. Аналіз платформи Home Assistant як основи для реалізації системи

Home Assistant є відкритою платформою автоматизації, яка дозволяє інтегрувати різні пристрої, сервіси та сценарії в єдиному інформаційному середовищі. Для цієї бакалаврської роботи платформа є цінною тим, що дає змогу реалізувати одразу кілька рівнів системи: серверний, інтеграційний, інтерфейсний і частково аналітичний.

Переваги Home Assistant у задачах Smart Campus

До переваг Home Assistant для задач Smart Campus належать підтримка великої кількості інтеграцій, наявність веб-інтерфейсу, журналу подій, історії, автоматизацій, а також можливість підключення ESPHome.

Особливо важливо, що платформа дозволяє відображати реальні пристрої, їхні сутності, історію та автоматизації. Це робить її зручною для бакалаврського проєкту, де цінується не лише теоретична модель, а й можливість практичної демонстрації.

Обмеження Home Assistant

Попри значні переваги, Home Assistant не є вузькоспеціалізованою платформою саме для систем контролю доступу. Це означає, що окремі механізми, характерні для промислових систем, у ньому доводиться проектувати й налаштовувати самостійно.

До обмежень можна віднести відсутність готової коробкової логіки доступу, потребу в додатковому проектуванні структури БД, залежність від коректності конфігурації інтеграцій та необхідність самостійно визначати ролі, зони й правила.

На основі проведеного аналізу задачу бакалаврської роботи можна сформулювати як створення підсистеми, що забезпечує ідентифікацію користувача за RFID/NFC-міткою, перевіряє відповідність користувача і його ролі встановленим правилам доступу, реєструє всі події проходу, формує інформацію для моніторингу й аналізу, інтегрується в середовище Home Assistant і допускає подальше масштабування.

Вхідна та вихідна інформація системи

До вхідної інформації системи належать UID мітки, дані про користувача, ролі користувачів, зони доступу, точки проходу, правила доступу, статуси міток і часові параметри.

До вихідної інформації належать рішення про доступ, запис у журналі подій, інформація в інтерфейсі Home Assistant, повідомлення про інцидент та звітні й статистичні дані.

Основні задачі системи

Для досягнення мети система повинна вирішувати такі задачі: зчитувати мітку на точці доступу, встановлювати зв’язок між міткою і користувачем, перевіряти права доступу, фіксувати результат проходу, передавати дані в інтерфейс користувача та забезпечувати подальший аналіз подій.

Таким чином, постановка задачі охоплює не лише технічну організацію проходу, а й створення повноцінної інформаційної підсистеми.

При розгляді можливостей системи контролю відвідувачів доцільно перейти від загальних теоретичних положень до аналізу типових практичних сценаріїв, характерних саме для університетського середовища. Це дає змогу уточнити вимоги до системи та зрозуміти, як вона буде поводитися в реальних умовах експлуатації.

Першим типовим сценарієм є доступ студентів до навчальних корпусів і аудиторій у межах встановленого розкладу. Для цієї категорії користувачів система повинна забезпечувати швидку ідентифікацію, мінімізувати затримки на вході та водночас враховувати часові й просторові обмеження.

Другим сценарієм є доступ працівників до службових кабінетів, лабораторій і спеціалізованих приміщень. У цьому випадку важливо враховувати не лише належність до ролі, а й специфіку робочих обов’язків та закріплених зон.

Окремим сценарієм є доступ відвідувачів або гостей. Для такої категорії система має підтримувати тимчасові або обмежені правила проходу, що особливо важливо під час проведення конференцій, відкритих лекцій або зовнішніх заходів.

Таким чином, аналіз практичних сценаріїв демонструє, що система контролю відвідувачів у межах університету повинна бути гнучкою, диференційованою за ролями та придатною до роботи як у типових, так і в нештатних ситуаціях.

Висновки до розділу 1

У першому розділі було проаналізовано предметну область бакалаврської роботи та розглянуто роль підсистеми контролю відвідувачів у структурі Smart Campus. Встановлено, що контроль доступу є важливою складовою безпекової інфраструктури сучасного закладу освіти та має розглядатися як інформаційна, а не лише технічна система.

Було проведено аналіз існуючих технологій ідентифікації та показано, що найбільш доцільним для цього проєкту є RFID/NFC-підхід. Обґрунтовано

використання RC522 для демонстраційної реалізації та PN532 як рекомендованого рішення для перспективного впровадження.

Також доведено, що Home Assistant є доцільною основою для реалізації підсистеми контролю відвідувачів, оскільки поєднує інтеграційні, серверні, інтерфейсні та моніторингові можливості. На завершення сформульовано постановку задачі розробки системи та визначено її вхідну і вихідну інформацію.

РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ КОНТРОЛЮ ВІДВІДУВАЧІВ

2.1. Загальна концепція та функціональна структура системи

Проєктування інформаційної системи контролю відвідувачів для Smart Campus повинно базуватися на поєднанні функціональної повноти, технічної реалізовуваності, інтегрованості в цифрове середовище та можливості подальшого розширення.

Загальна концепція системи полягає в тому, що доступ до певної точки проходу надається або блокується на основі перевірки сукупності умов: існування мітки в системі, прив’язки мітки до користувача, ролі користувача, відповідності точки проходу заданій зоні, часових обмежень та активності правил доступу.

Функціональна структура системи включає апаратний, інтеграційний, логічний, інформаційний та інтерфейсний рівні.

Формування вимог є необхідним етапом проектування, оскільки саме вимоги визначають, які функції повинна забезпечувати система, які обмеження мають бути враховані та якими критеріями оцінюватиметься повнота її реалізації.

Функціональні вимоги до системи

Система повинна забезпечувати зчитування RFID або NFC-мітки на точці доступу, визначати унікальний ідентифікатор мітки, перевіряти наявність мітки в системі, визначати користувача та його роль, перевіряти правила доступу за роллю, зоною та часовим інтервалом, приймати рішення про надання або відмову в доступі, реєструвати всі події проходу, формувати повідомлення про інцидент, забезпечувати перегляд журналу подій і підтримувати підготовку даних для звітів та статистики.

Нефункціональні вимоги до системи

Система повинна бути інтегрованою в середовище Home Assistant, підтримувати масштабування на кілька точок доступу, мати зрозумілий і доступний інтерфейс користувача, забезпечувати централізоване збереження подій, бути придатною до модернізації та допускати перехід від RFID до NFC без повної перебудови архітектури.

2.2. Моделювання функціонування системи засобами UML

Для формалізації структури й поведінки системи доцільно використовувати UML-моделювання. У межах даної роботи застосовуються діаграма прецедентів, діаграма послідовності та діаграма діяльності.

Діаграма прецедентів відображає функції системи з погляду зовнішніх користувачів. Основними акторами є відвідувач, адміністратор системи та охорона.

UML-діаграма прецедентів системи контролю відвідувачів наведена на рис. 2.1.

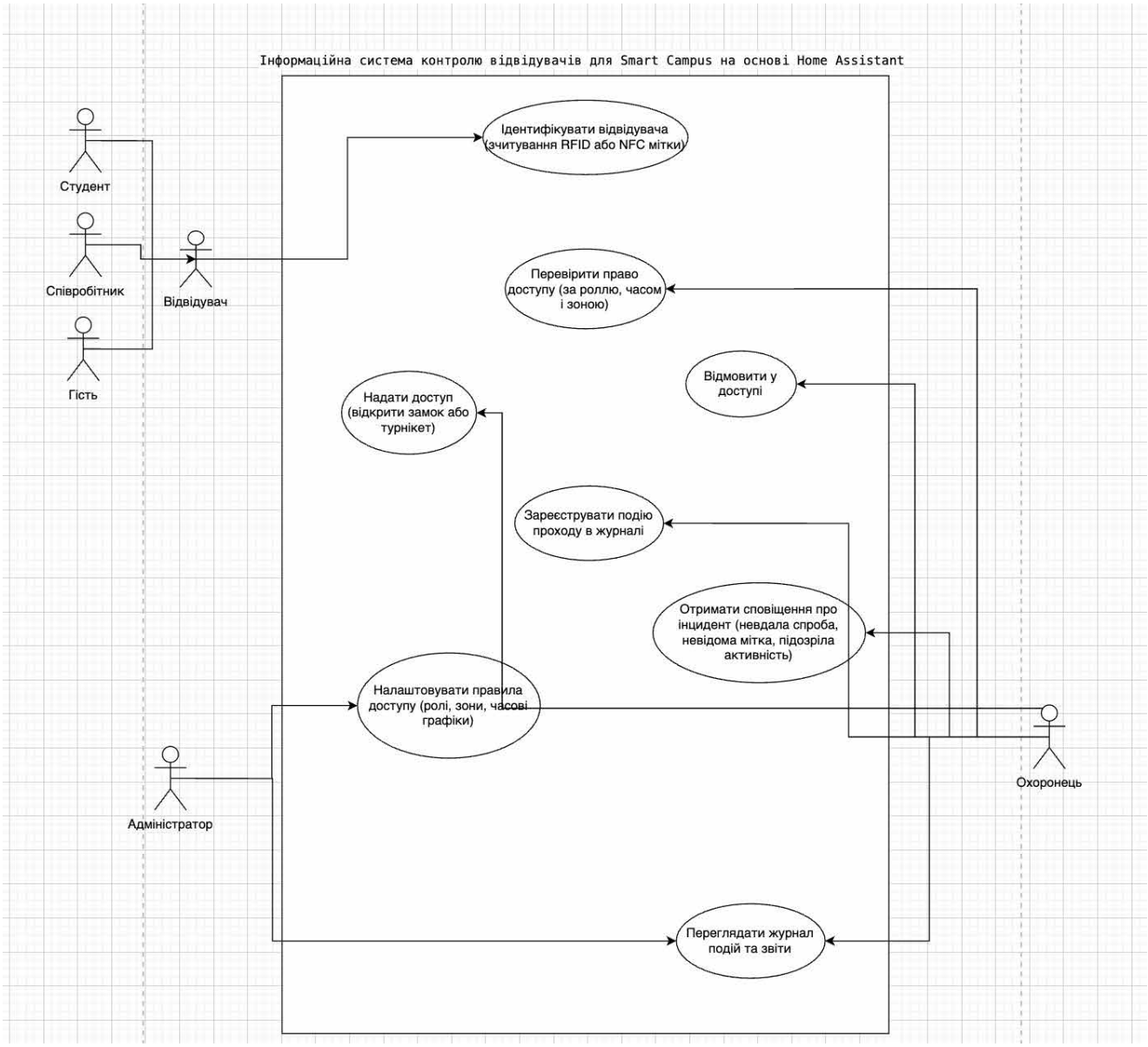

Рис. 2.1. UML-діаграма прецедентів системи контролю відвідувачів

Діаграма послідовності описує взаємодію між компонентами системи в часі та показує, як подія проходу проходить шлях від фізичного зчитування до логічного рішення системи. UML-діаграма послідовності системи контролю відвідувачів наведена на рис. 2.2.

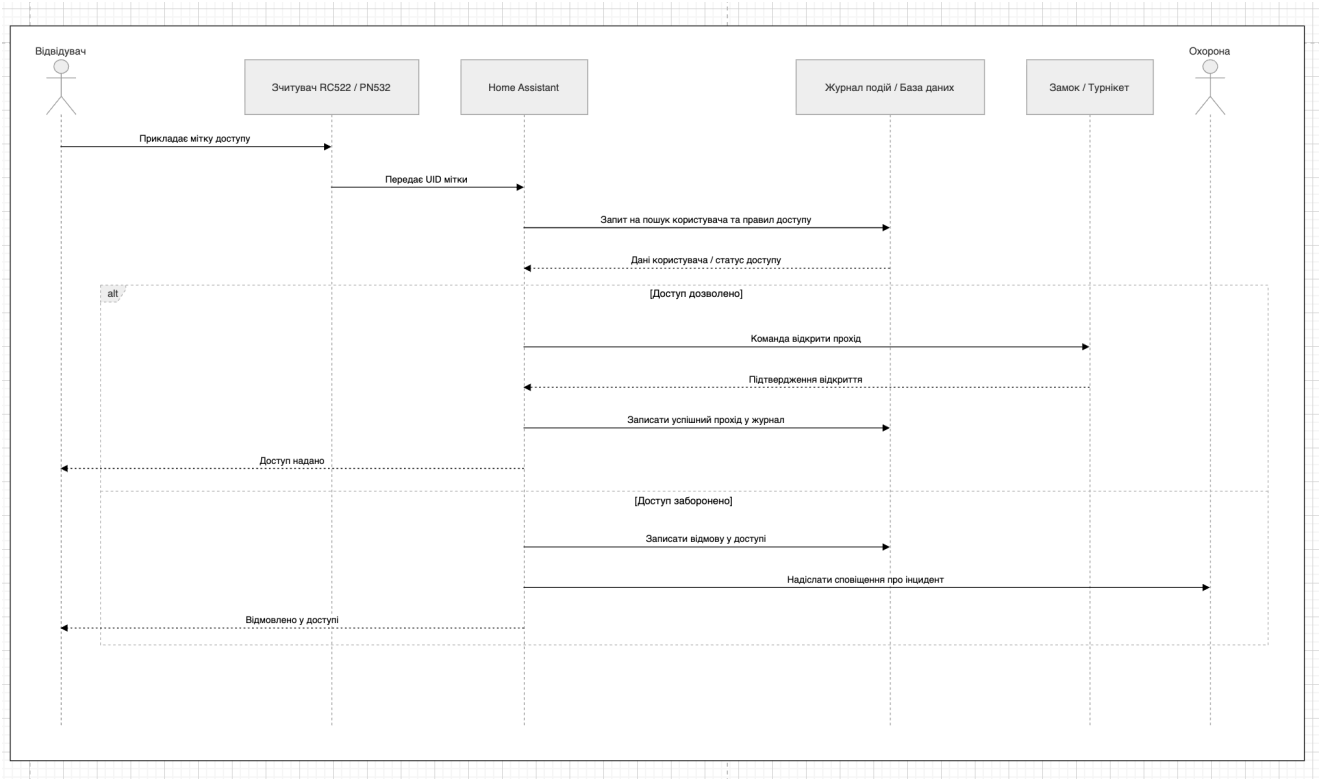

Рис. 2.2. UML-діаграма послідовності системи контролю відвідувачів

Діаграма діяльності відображає алгоритм функціонування системи у вигляді послідовності дій і розгалужень. Вона використовується для опису логіки перевірки доступу та сценаріїв успішного чи неуспішного проходу. UML-діаграма діяльності системи контролю відвідувачів наведена на рис. 2.3

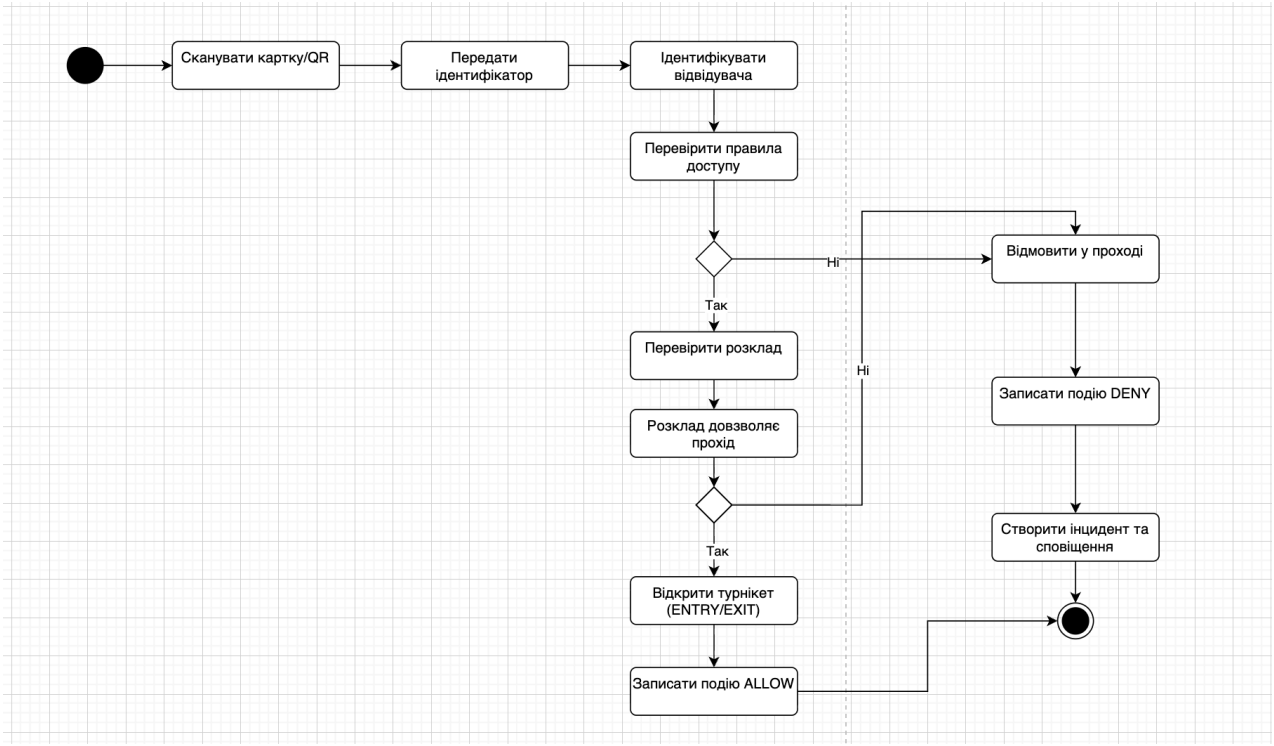

Рис. 2.3. UML-діаграма діяльності системи контролю відвідувачів

Застосування UML дозволило узгодити функціональні вимоги з логікою системи, а також заздалегідь описати ключові точки взаємодії між користувачем, точкою доступу, Home Assistant та базою даних.

2.3. Проєктування сценаріїв взаємодії користувача і системи

Сценарій проходу зареєстрованого користувача

У цьому сценарії користувач прикладає мітку до зчитувача, система отримує UID мітки, виконує пошук мітки, визначає користувача, перевіряє його роль, перевіряє правила доступу та, в разі позитивного результату, дозволяє прохід і записує подію як успішну.

Сценарій відмови в доступі

У даному сценарії можливі кілька причин негативного результату: мітка не знайдена, мітка неактивна, користувач не має права доступу до зони або доступ у даний час заборонений. У такому випадку система фіксує відмову, зберігає

причину, створює запис у журналі та, за потреби, формує інцидентне повідомлення.

Сценарій перегляду журналу подій

Цей сценарій стосується адміністратора або охорони. Користувач відкриває журнал подій або історію та аналізує зафіксовані записи, не звертаючись безпосередньо до бази даних.

Сценарій технічного моніторингу вузла

У цьому сценарії адміністратор працює з ESPHome Builder, переглядає вузол ESPHome RFID#1, його статус, журнали, конфігурацію або пов’язані сутності.

Архітектура системи має модульний характер. Основними модулями є інтерфейс користувача, модуль контролю доступу, модуль журналювання подій, модуль сповіщень, модуль доступу до бази даних, інтеграція з Home Assistant, інтеграція зі зчитувачем, модуль конфігурації та модуль формування звітів. Центральним елементом архітектури є модуль контролю доступу.

Модульний підхід дозволяє розділити систему на окремі функціональні частини, кожна з яких виконує власну задачу. Це спрощує аналіз, реалізацію, тестування та подальший розвиток системи. Крім того, у модульній системі простіше виконувати модернізацію: заміна RC522 на PN532 або розширення звітного модуля не вимагає перебудови всієї архітектури.

Для глибшого розуміння побудованої архітектури доцільно розглянути декомпозицію функцій системи та основні інформаційні потоки, що циркулюють між її компонентами. Такий аналіз дозволяє побачити систему не лише як набір модулів, а як послідовність взаємопов’язаних процесів.

На вхід системи надходить подія зчитування мітки. Первинним джерелом цієї події є апаратний вузол на базі ESP32, підключений до RFID-зчитувача. Після реєстрації UID подія передається через ESPHome до Home Assistant.

Другий інформаційний потік пов’язаний із доступом до довідкових даних про користувачів, ролі, мітки, зони доступу та правила проходу. Саме цей потік забезпечує інтерпретацію події в контексті всієї системи.

Третій потік стосується формування результату. Після перевірки рольової моделі та часових обмежень система видає рішення про надання або відмову в доступі та запускає журналювання події.

Четвертий потік пов’язаний із представленням інформації користувачу. Результати подій відображаються в Home Assistant через сторінки пристроїв, журнал, історію та інформаційні панелі.

Висновки до розділу 2

У другому розділі сформовано загальну концепцію та функціональну структуру системи, яка включає апаратний, інтеграційний, логічний, інформаційний та інтерфейсний рівні.

Було визначено функціональні та нефункціональні вимоги до системи, виконано UML-моделювання функціонування та розглянуто основні сценарії взаємодії користувачів із нею.

Окрему увагу приділено проектуванню архітектури програмного забезпечення. Обґрунтовано використання модульного підходу, який забезпечує гнучкість, масштабованість і зручність подальшого розвитку системи.

РОЗДІЛ 3 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ БАЗИ ТА ПРОГРАМНО-АПАРАТНОЇ АРХІТЕКТУРИ

3.1. Проєктування логічної моделі даних

Одним із ключових етапів розробки інформаційної системи контролю відвідувачів є побудова логічної моделі даних. Саме логічна модель визначає, які інформаційні об’єкти повинні бути відображені в системі, якими атрибутами вони характеризуються та які взаємозв’язки існують між ними. Для підсистеми контролю доступу це має особливе значення, оскільки система одночасно працює з користувачами, мітками, ролями, зонами, точками проходу, подіями та правилами доступу.

Логічна модель у межах цієї роботи формується на основі функціональних вимог до системи. Оскільки підсистема повинна підтримувати ідентифікацію відвідувачів, перевірку прав доступу, журналювання подій і формування повідомлень про інциденти, набір сутностей не може обмежуватись лише таблицею користувачів і таблицею міток. Необхідно забезпечити таку структуру, яка дозволить прив’язати мітку до конкретного користувача, прив’язати користувача до певної ролі, визначити, до яких зон має доступ та чи інша роль, зафіксувати, на якій точці проходу сталася подія, зберегти результат перевірки доступу та виділити інцидентні ситуації в окремий інформаційний контур.

На основі цього було виділено вісім основних сутностей: ROLE, VISITOR, ACCESS_TAG, ACCESS_ZONE, ACCESS_POINT, ACCESS_RULE, ACCESS_EVENT, SECURITY_NOTIFICATION.

Сутність ROLE призначена для опису ролей користувачів у системі. Використання ролей дає змогу побудувати гнучкий механізм розмежування доступу. У межах Smart Campus це особливо важливо, оскільки правила для студентів, працівників, охорони, адміністраторів та тимчасових відвідувачів

суттєво відрізняються. До складу атрибутів цієї сутності входять ідентифікатор ролі, назва ролі та її опис.

Сутність VISITOR є центральною з погляду персоніфікації доступу. Вона містить дані про конкретного користувача системи: ідентифікатор, ПІБ, тип користувача, контактну інформацію, статус, а також посилання на роль. Зв’язок із роллю дозволяє уникнути дублювання правил доступу для кожного окремого запису користувача.

Сутність ACCESS_TAG використовується для опису міток доступу. У системі одна мітка має бути прив’язана до одного користувача, але один користувач потенційно може мати кілька міток, якщо це потрібно з організаційних причин. У цій сутності зберігається UID мітки, тип мітки, дата видачі, строк дії та поточний статус.

Сутність ACCESS_ZONE відображає логічне групування простору. Наприклад, у навчальному корпусі можуть бути окремі зони: головний вхід, навчальні аудиторії, лабораторії, службові приміщення. Саме через цю сутність система отримує можливість працювати не лише з окремими дверима чи турнікетами, а з цілими функціональними ділянками.

Сутність ACCESS_POINT конкретизує фізичне місце проходу. На відміну від зони, яка має логічний характер, точка проходу є технічною сутністю, яка відповідає конкретному зчитувачу, дверям, турнікету або виконавчому вузлу.

Сутність ACCESS_RULE реалізує логіку доступу. У ній поєднуються роль користувача, зона доступу, дні тижня, часові обмеження та ознака активності правила. Саме ця сутність фактично визначає, хто, куди і коли має право проходу.

Сутність ACCESS_EVENT є журналом усіх спроб доступу. Тут зберігається інформація про дату й час події, користувача, мітку, точку проходу, результат перевірки та причину відмови. Це одна з найважливіших сутностей системи, оскільки вона слугує основою для звітності, статистики та аналізу інцидентів.

Сутність SECURITY_NOTIFICATION використовується для виділення інцидентних подій у самостійний інформаційний потік. Вона дає можливість не лише фіксувати факт відмови, а й відокремлювати події, які потребують уваги охорони або адміністратора.

Отже, логічна модель даних дозволяє відобразити всі ключові інформаційні об’єкти системи та підготувати основу для переходу до фізичного рівня реалізації бази даних.

Для більш повного обґрунтування побудованої моделі доцільно детальніше розглянути характер зв’язків між основними сутностями.

Між сутностями ROLE та VISITOR існує зв’язок типу один-до-багатьох. Одна роль може бути призначена багатьом користувачам, але кожен користувач у базовій моделі належить лише до однієї ролі. Такий підхід є раціональним, оскільки для задачі контролю доступу основна логіка зазвичай визначається саме за роллю.

Між VISITOR та ACCESS_TAG також реалізується зв’язок один-до-багатьох. Це означає, що користувач може мати одну або кілька міток доступу. У практичному сенсі це дозволяє враховувати сценарії перевидачі мітки, тимчасового дублювання або використання різних засобів ідентифікації.

Між ACCESS_ZONE та ACCESS_POINT існує зв’язок один-до-багатьох, оскільки одна зона може містити кілька точок проходу. Це важливо з архітектурного погляду, адже правила доступу зазвичай задаються на рівні зони, а фактичні події фіксуються на рівні конкретної точки проходу.

Сутність ACCESS_RULE пов’язує між собою ROLE та ACCESS_ZONE. У такий спосіб створюється гнучкий механізм призначення прав доступу. Замість індивідуального налаштування доступу для кожного користувача система працює через узагальнений механізм ролей і зон.

Між VISITOR, ACCESS_TAG, ACCESS_POINT та ACCESS_EVENT також реалізуються зв’язки один-до-багатьох. Кожен користувач, кожна мітка і кожна

точка проходу можуть фігурувати в багатьох подіях. Це є природним, оскільки журнал подій формується безперервно в процесі функціонування системи.

Сутність SECURITY_NOTIFICATION пов’язана з ACCESS_EVENT, оскільки інцидентне повідомлення створюється на основі конкретної події доступу. У базовій логіці можна вважати, що одна подія може породити одне або кілька службових повідомлень залежно від рівня деталізації та складності сценарію реагування.

З точки зору нормалізації така модель є доцільною, оскільки зменшує дублювання даних, дозволяє централізовано змінювати ролі, правила і зони, спрощує побудову звітів та забезпечує логічну прозорість структури системи.

Таким чином, побудована логічна модель є цілісною, структурованою та відповідає функціональним задачам підсистеми контролю відвідувачів.

3.2. Проєктування фізичної моделі бази даних

На фізичному рівні структура бази даних повинна бути не лише формально коректною, а й зручною для подальшої експлуатації. Саме тому для кожної таблиці доцільно визначати не тільки перелік полів, а й їх прикладне призначення, спосіб використання та очікувані сценарії оновлення. Фізична модель бази даних системи наведена на рис. 3.1. Наприклад, таблиця ролей змінюється порівняно рідко, оскільки набір ролей у реальному середовищі університету є відносно стабільним. Натомість таблиця подій проходу поповнюється постійно, адже кожна спроба доступу породжує новий запис. Це означає, що до різних таблиць висуваються різні вимоги щодо частоти читання, запису та подальшої індексації.

Для таблиці користувачів важливо передбачити поля, які дозволяють не лише ідентифікувати особу, а й надалі використовувати ці дані для адміністративних дій. До таких полів належать повне ім’я, ознака активності, роль, тип користувача та контактні атрибути. Якщо в майбутньому система буде розширена, до цієї таблиці

можуть додаватися дані про належність до підрозділу, навчальної групи, факультету або категорії персоналу. Однак уже в межах поточного проєкту важливо закласти сам принцип розширюваності структури.

Для таблиці міток суттєвим є збереження унікального ідентифікатора, типу мітки, стану, дати видачі та строку дії. Така структура дозволяє гнучко керувати життєвим циклом мітки: створенням, активацією, тимчасовим блокуванням, перевипуском та остаточним відкликанням. Це особливо важливо для навчального закладу, де склад користувачів та їх статус може досить часто змінюватися у зв'язку із завершенням навчання, переведенням, відпустками чи тимчасовими дозволами на доступ.

Для таблиць зон та точок проходу доцільно розділяти логічний і фізичний рівні опису простору. Зона є зручним поняттям для призначення правил доступу, тоді як точка проходу фіксує конкретне місце взаємодії користувача із системою. Такий підхід дає змогу пов'язати кілька фізичних входів з однією функціональною зоною, а також спрощує майбутнє масштабування. Наприклад, якщо до певної лабораторії ведуть двоє дверей, то система може мати дві точки проходу в межах однієї й тієї самої зони.

Окремої уваги потребує таблиця правил доступу. Її структура повинна дозволяти описувати доступ як у простій формі, так і в більш гнучкому вигляді. У базовому варіанті правило може задавати роль, зону та часовий інтервал. Проте вже на етапі проєктування доцільно враховувати, що в перспективі правила можуть ускладнюватися: з’явиться потреба у винятках, пріоритетах, індивідуальних дозволах для окремих користувачів чи тимчасових перепустках. Тому навіть навчальна модель має бути побудована без зайвої жорсткості.

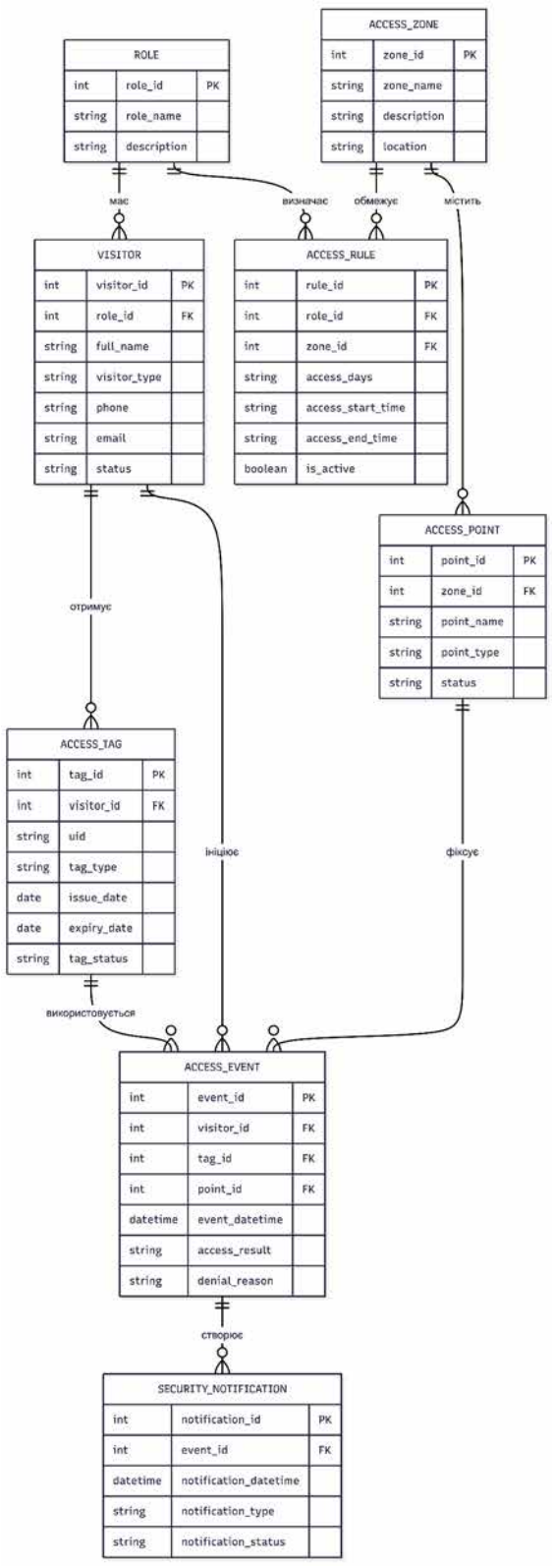

Рис. 3.1. Фізична модель бази даних системи

Найбільш динамічною таблицею є журнал подій. Саме він накопичує реальну історію роботи системи, а отже з часом може стати найбільшою таблицею за обсягом. Через це її структура має бути достатньо простою для швидкого запису та подальшої вибірки. Крім базових полів, тут доцільно передбачити текстову причину відмови та ознаку, яка дозволяє відокремити звичайну помилку від потенційно небезпечного інциденту.

Вибір типів даних для полів бази має не лише технічне, а й прикладне значення. Наприклад, для первинних ключів доцільно використовувати цілі числові типи, оскільки вони забезпечують зручне індексування та швидке з’єднання таблиць. Для текстових полів, таких як назва ролі чи повне ім'я користувача, доцільно застосовувати рядкові типи із розумним обмеженням довжини, що дозволяє уникати надмірного використання пам'яті та робить структуру більш контрольованою.

Для часових параметрів потрібно використовувати окремі типи дати та часу, а не зберігати такі значення у вигляді тексту. Це забезпечує коректне сортування, порівняння та побудову запитів за часовими інтервалами. Особливо це важливо для таблиці подій проходу та таблиці правил доступу, де дата й час є основою логіки функціонування системи.

Первинні ключі в системі мають забезпечувати однозначну ідентифікацію записів, а зовнішні ключі − цілісність зв’язків між таблицями. Для запобігання дублюванню записів окремі поля, наприклад UID мітки чи назва ролі, повинні мати ознаку унікальності. Такий підхід дає змогу не покладатися лише на прикладну логіку, а забезпечити частину контролю безпосередньо на рівні СУБД.

Індексування в цій системі слід розглядати не як формальне доповнення, а як інструмент оптимізації типових операцій. До полів, які доцільно індексувати, належать зовнішні ключі, UID мітки, дата й час події, а також поля, які найчастіше використовуються у фільтрації звітів. За рахунок цього пришвидшується пошук

користувача за міткою, вибірка подій за період, аналіз подій по окремих точках проходу та формування статистичних зведень.

Практична реалізація структури бази даних передбачає створення таблиць, індексів, обмежень цілісності, а також допоміжних SQL-об'єктів, які спрощують роботу із системою. До останніх належать представлення, процедури, тригери та службові запити початкового заповнення. Навіть якщо в межах бакалаврської роботи реалізується не весь набір таких елементів, сама структура проєкту повинна бути підготовлена так, щоб їх легко можна було додати без переробки базової схеми.

Наприклад, процедура реєстрації події проходу може приймати як параметри ідентифікатор мітки, точку проходу, результат перевірки та причину відмови. Така процедура уніфікує процес запису події й зменшує ризик різночитань у різних частинах програмної логіки. Аналогічно, окремий тригер може автоматично створювати запис про інцидент, якщо подія доступу завершилася відмовою з певною категорією причин.

Допоміжні представлення доцільно використовувати для звітних задач. Наприклад, окреме представлення може містити вже з'єднані дані про подію, користувача, роль та точку проходу. Це дозволяє значно спростити побудову аналітичних запитів і дає змогу відокремити операційний рівень БД від рівня звітної аналітики.

Для повторного розгортання системи доцільно підготувати інсталяційний пакет бази даних у вигляді логічно впорядкованого набору SQL-файлів. Такий пакет має містити скрипти створення таблиць, індексів, допоміжних процедур, тригерів і початкових довідкових записів. Важливо, щоб порядок виконання цих файлів був однозначним і не викликав конфліктів через відсутність пов'язаних таблиць або об’єктів.

У практичному сенсі це означає, що розгортання бази даних має виконуватися не вручну й не хаотично, а за підготовленим сценарієм. Такий підхід особливо цінний для навчального проєкту, оскільки демонструє завершеність інженерної логіки: від моделі та схеми − до готового набору матеріалів, придатних для інсталяції й повторного запуску в іншому середовищі.

База даних у розроблюваній системі не є ізольованим компонентом, що існує сам по собі. Вона тісно пов'язана із серверною логікою, яка виконує перевірку доступу, журналювання та підготовку даних для відображення в Home Assistant. Це означає, що база повинна бути спроєктована так, щоб її структура була зручною для інтеграції із серверним рівнем і не створювала зайвих труднощів під час побудови запитів.

Для взаємодії із серверною частиною особливо важливими є два класи операцій. Перший клас − це короткі операційні запити, які виконуються під час обробки події доступу: пошук мітки, визначення користувача, перевірка ролі та активних правил. Другий клас − це аналітичні або оглядові запити, які використовуються для журналів, звітів і статистичних зведень. Саме тому структура БД повинна одночасно бути придатною і для швидкої транзакційної роботи, і для побудови узагальнених вибірок.

Після визначення сутностей і зв’язків між ними логічна модель трансформується у фізичну модель бази даних. На цьому етапі кожна сутність набуває форми конкретної таблиці, а її атрибути — форми полів з визначеними типами даних.

3.3. Програмно-апаратна структура системи

Структурна схема системи контролю відвідувачів Smart Campus наведена на рис. 3.2.

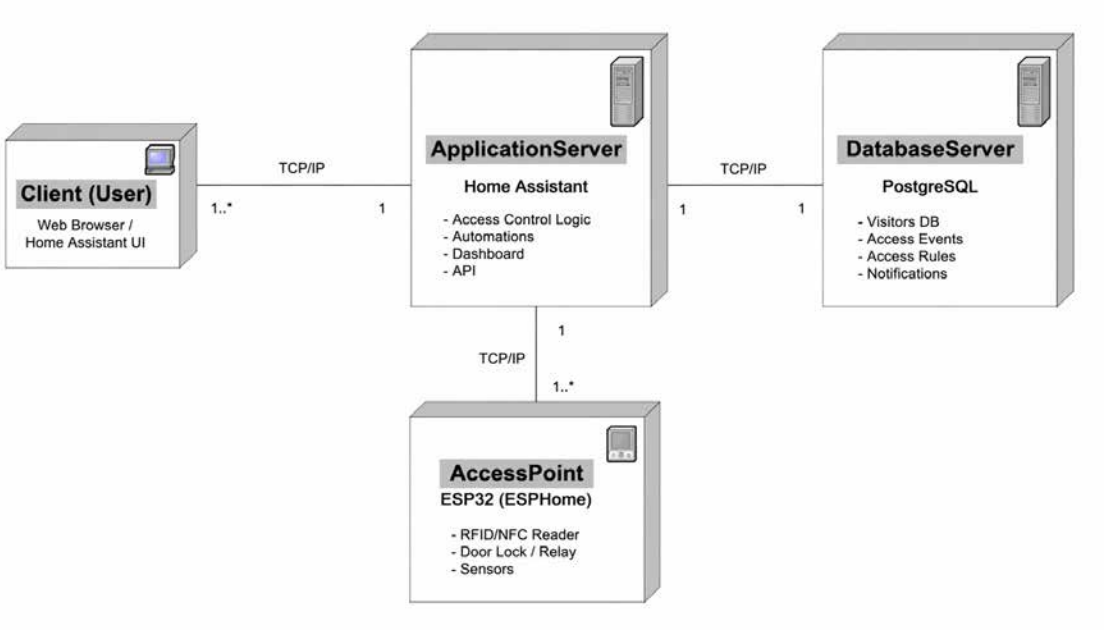

Рис. 3.2. Структурна схема системи контролю відвідувачів Smart Campus

Програмно-апаратна структура системи включає серверну частину на базі Home Assistant, базу даних PostgreSQL, вузол доступу на базі ESP32, інтеграційний прошарок ESPHome, RFID/NFC-зчитувач, виконавчий пристрій та клієнтський веб-інтерфейс.

Серверна частина відповідає за обробку подій, автоматизації та інтерфейсне представлення. База даних виконує роль централізованого сховища структурованої інформації. Апаратний вузол забезпечує фізичний контакт із користувачем через мітку.

У демонстраційному макеті використовується RC522. Однак у роботі слід наголошувати, що для реального впровадження рекомендованим є PN532, оскільки він краще підтримує NFC-сценарії.

Окрім визначення сутностей і зв’язків, важливо оцінити побудовану структуру бази даних з позицій нормалізації та забезпечення цілісності. Для системи контролю доступу це особливо суттєво, оскільки помилки в організації даних можуть призвести до дублювання ролей, суперечностей у правилах доступу або некоректного трактування подій проходу.

Побудована модель відповідає загальним принципам нормалізованого проєктування. Довідкові сутності, такі як ролі, зони доступу та точки проходу, винесені в окремі таблиці. Це дозволяє уникнути дублювання текстових значень у таблицях подій та правил.

Цілісність даних забезпечується використанням первинних і зовнішніх ключів. Наявність зовнішніх ключів між visitors, access_tags, access_rules, access_points та access_events гарантує, що подія проходу не може посилатися на неіснуючого користувача або неіснуючу точку.

Окремої уваги потребує оптимізація запитів до таблиці access_events, яка потенційно буде найбільшою за обсягом. Для пришвидшення вибірки доцільно створювати індекси за полями event_datetime, visitor_id, point_id та access_result.

Таким чином, побудована база даних є не лише формально повною, а й достатньо стійкою до дублювання, логічних суперечностей і типових помилок адміністрування.

Висновки до розділу 3

У третьому розділі було розроблено логічну та фізичну моделі бази даних системи контролю відвідувачів для Smart Campus. Визначено основні сутності, встановлено зв’язки між ними та сформовано фізичну структуру таблиць.

Було обґрунтовано вибір PostgreSQL як основної СУБД системи та показано її придатність для реалізації реляційної моделі з підтримкою ключів, індексів, процедур і тригерів.

Також було описано підхід до реалізації SQL-об’єктів та формування інсталяційного пакета бази даних. Окремо визначено програмно-апаратну структуру системи, яка поєднує Home Assistant, PostgreSQL, ESPHome, ESP32 та RFID/NFC-зчитування.

РОЗДІЛ 4 РЕАЛІЗАЦІЯ ПІДСИСТЕМИ КОНТРОЛЮ ВІДВІДУВАЧІВ НА БАЗІ HOME ASSISTANT

4.1. Реалізація серверної частини системи в середовищі Home Assistant

Практична реалізація підсистеми контролю відвідувачів у межах даної бакалаврської роботи здійснюється на базі платформи Home Assistant, яка виступає центральним програмним середовищем Smart Campus. Саме в цій платформі зосереджено інтеграцію апаратних вузлів, логіку автоматизацій, інтерфейс користувача, відображення станів, журнал подій та історію роботи системи. Панель «Огляд» із картками стану системи наведена на рис. 4.1.

Рис. 4.1. Панель «Огляд» із картками стану системи

Використання Home Assistant як серверної основи є доцільним з кількох причин. По-перше, дана платформа вже орієнтована на інтеграцію різнорідних пристроїв і сервісів у межах одного середовища. По-друге, вона має готовий веб-

інтерфейс, що дозволяє реалізувати інтерфейсну частину без розробки окремого клієнтського застосунку. По-третє, Home Assistant підтримує автоматизації, які дають змогу організувати реакцію системи на події, що надходять від вузла доступу.

У контексті цієї роботи серверна частина системи виконує такі функції: приймає події від апаратного вузла доступу, забезпечує їх інтерпретацію в логіці системи, відображає стани пристроїв, зберігає й передає інформацію до механізмів журналювання, забезпечує взаємодію з базою даних та формує інтерфейс моніторингу й керування.

У середовищі Home Assistant доступними є такі компоненти, що мають значення для даного проєкту: Dashboard, ESPHome Builder, Devices & Services, Automations, History, Logbook. Важливою перевагою такого підходу є відсутність потреби у створенні окремої багаторівневої серверної архітектури.

Таким чином, серверна частина системи в межах даної роботи реалізована як реально функціонуюче середовище Home Assistant, доступ до якого вже організований і яке використовується для практичної демонстрації підсистеми контролю відвідувачів.

ESPHome Builder зі списком доступних вузлів наведена на рис. 4.2.

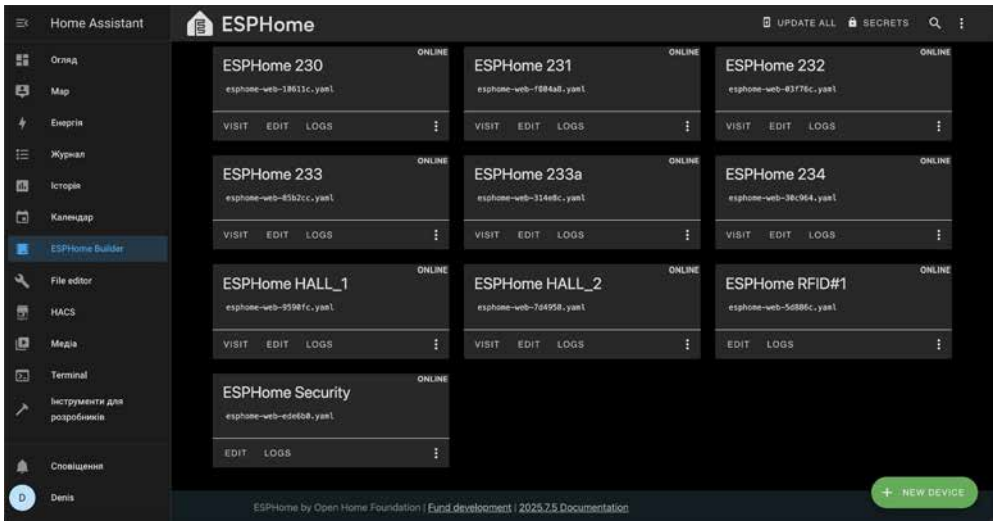

Рис. 4.2. Сторінка ESPHome Builder зі списком вузлів

Список інтеграцій у розділі Devices & Services наведений на рис. 4.3.

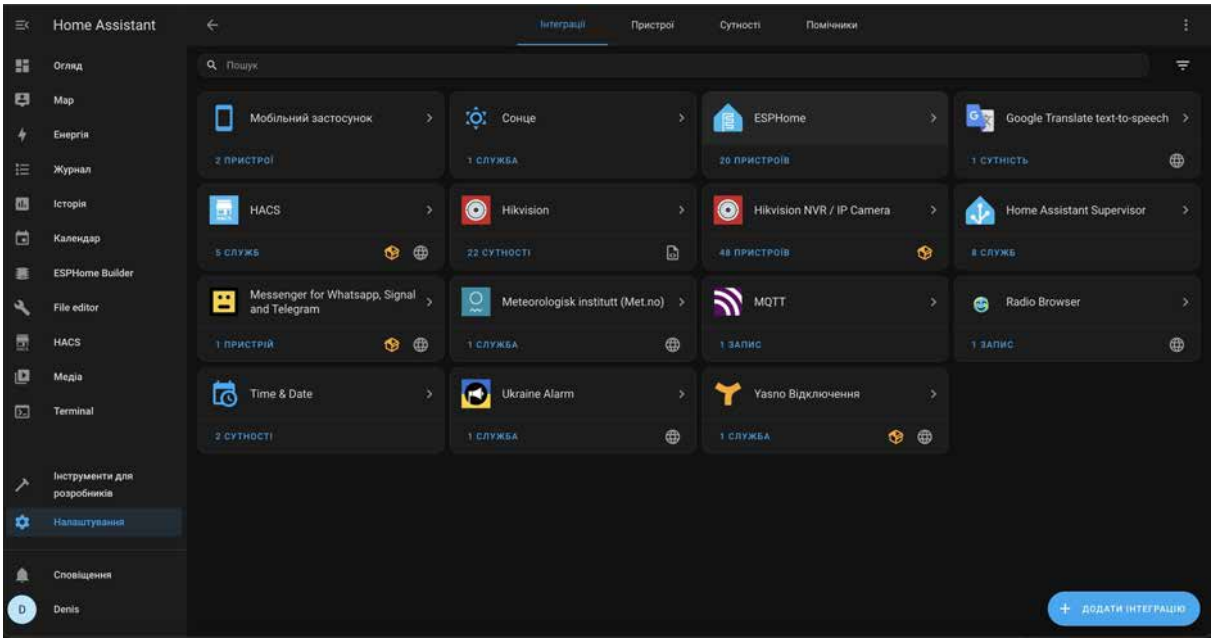

Рис. 4.3. Перелік інтеграцій у Home Assistant

Для впровадження підсистеми контролю відвідувачів на базі Home Assistant необхідно виконати послідовність технічних дій, пов’язаних із розгортанням серверного середовища, підключенням апаратного вузла та перевіркою коректності роботи всіх складових.

Першим етапом є розгортання Home Assistant. Серверна частина може бути встановлена на міні-ПК, окремому комп’ютері або іншому сумісному пристрої. Після встановлення платформи необхідно забезпечити доступ до веб-інтерфейсу, виконати первинне налаштування системи та переконатися, що основне середовище готове до інтеграції нових компонентів.

Другим етапом є налаштування базової структури Smart Campus у межах Home Assistant. Третім етапом є підключення ESPHome. Для цього в системі використовується ESPHome Builder, який дозволяє створювати конфігурацію вузла, завантажувати її на ESP32, перевіряти журнали роботи, редагувати YAML-файл конфігурації та контролювати стан з’єднання вузла із серверною частиною.

Четвертим етапом є налаштування вузла ESPHome RFID#1. Саме цей вузол відіграє роль апаратної точки доступу. На даному кроці потрібно задати ім’я пристрою, налаштувати мережеве підключення, визначити компоненти, які використовуються у вузлі, забезпечити інтеграцію з Home Assistant та перевірити, що після підключення вузол з’являється в системі як доступний пристрій.

П’ятим етапом є підключення апаратних компонентів до ESP32. Шостим етапом є налаштування автоматизацій. Сьомим етапом є перевірка інтерфейсного відображення. Восьмим етапом є тестова перевірка працездатності системи.

Таким чином, процес розгортання системи є поетапним і включає як серверну, так і апаратну частину. Для бакалаврської роботи важливо те, що всі ці етапи можуть бути продемонстровані через вже доступне середовище Home Assistant та ESPHome RFID#1.

4.2. Інтеграція ESPHome та вузла доступу ESPHome RFID#1

Журнали роботи вузла ESPHome RFID#1 наведений на рис. 4.4.

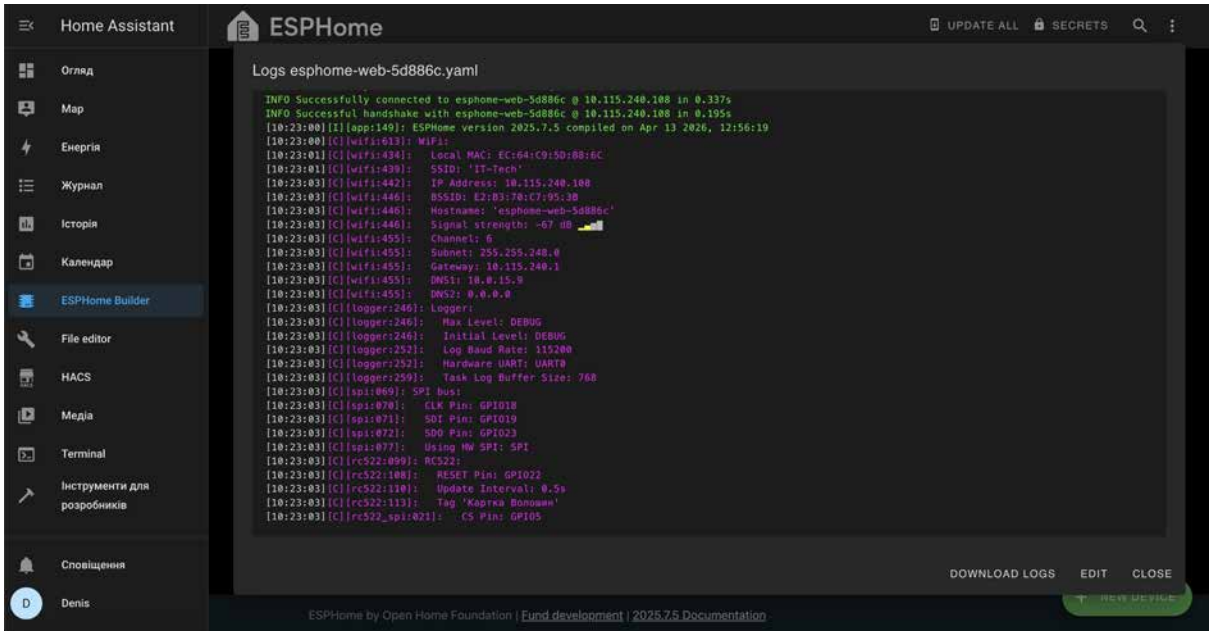

Рис. 4.4. Журнали роботи вузла ESPHome RFID#1

Перелік сутностей пристрою ESPHome RFID#1 наведена на рис. 4.5.

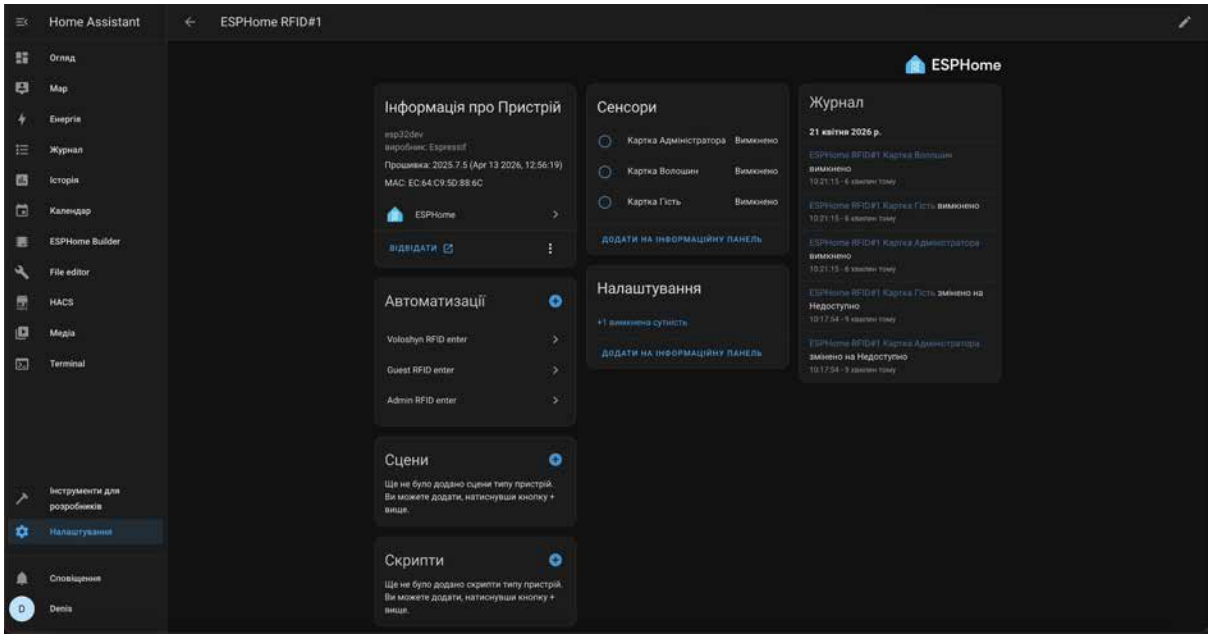

Рис. 4.5. Сутності вузла ESPHome RFID#1

Відкрита сторінка вузла ESPHome RFID#1 наведена на рис. 4.6.

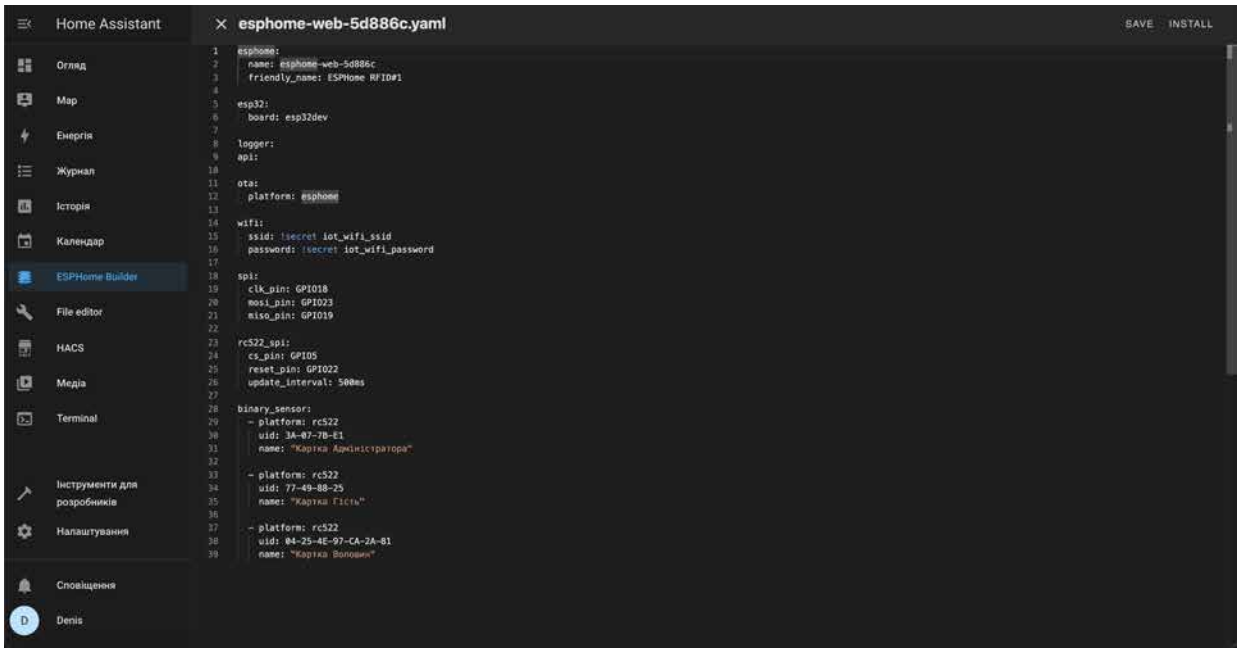

Рис. 4.6. Відкрита сторінка вузла ESPHome RFID#1

Ключовим елементом практичної реалізації підсистеми є вузол ESPHome RFID#1, який виконує роль апаратно-програмної точки доступу. Цей вузол побудований на базі мікроконтролера ESP32 і інтегрується з Home Assistant через ESPHome.

Основне призначення вузла полягає в тому, щоб зчитувати події від RFID-зчитувача, передавати їх у Home Assistant, брати участь у сценаріях контролю доступу та надавати системі технічну інформацію про власний стан.

Після інтеграції в Home Assistant вузол ESPHome RFID#1 стає доступним у кількох формах: як запис у списку пристроїв ESPHome Builder, як окремий пристрій у системі, як джерело пов’язаних сутностей та як джерело записів у журналах і технічних логах.

У межах поточної реалізації через ESPHome RFID#1 забезпечується зв’язок між фізичною подією зчитування та її цифровим відображенням у системі. Умовно цей процес можна подати так: мітка підноситься до зчитувача, зчитувач передає сигнал на ESP32, ESPHome обробляє цю подію як частину конфігурації вузла, інформація передається до Home Assistant, Home Assistant використовує її як подію або зміну стану, далі запускається логіка системи.

4.3. Реалізація логіки перевірки доступу, журналювання подій та відображення результатів

Центральною складовою підсистеми є логіка перевірки доступу. Саме вона визначає, чи отримає користувач доступ до певної зони, чи система сформує відмову.

У загальному вигляді логіка перевірки доступу включає такі етапи: отримання події про зчитування мітки, визначення ідентифікатора мітки, пошук мітки в системі, пошук користувача, якому належить мітка, визначення ролі

користувача, перевірка правил доступу, прийняття рішення про дозвіл або відмову, фіксація події в журналі та, за потреби, створення повідомлення про інцидент.

Сценарій успішного доступу

У разі успішного доступу система працює так: від користувача надходить фізична дія у вигляді прикладання мітки, система зчитує її ідентифікатор, визначається, що мітка є зареєстрованою, знаходиться відповідний користувач, перевіряється, чи має його роль право доступу до цієї зони в поточний момент часу, якщо всі умови виконані, формується позитивне рішення, подія проходу записується як успішна, інформація стає доступною для перегляду у Home Assistant.

Сценарій відмови в доступі

У разі неуспішного проходу можливі такі варіанти: мітка не знайдена в системі, мітка деактивована, користувач не має права доступу до заданої зони, правило доступу тимчасово неактивне, поточний час не входить у дозволений інтервал. У таких випадках система формує відмову в доступі, записує подію як неуспішну, за потреби ініціює інцидентне повідомлення та робить інформацію доступною в журналі та історії.

Історія змін станів пов’язаних сутностей наведена на рис. 4.7.

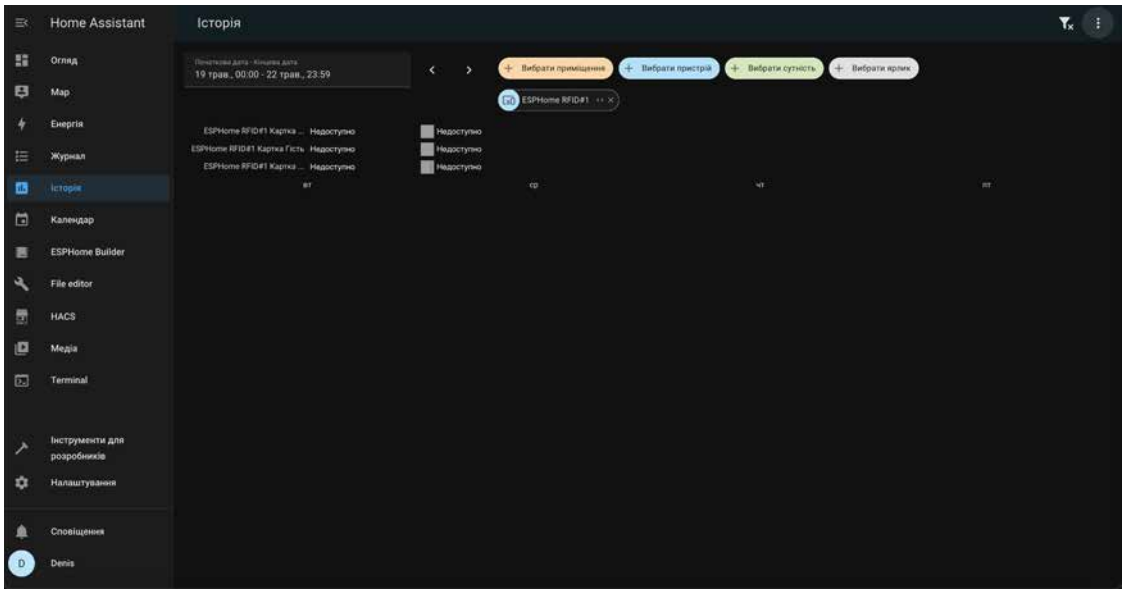

Рис. 4.7. Історія змін станів у Home Assistant

Журнал подій підсистеми контролю відвідувачів наведений на рис. 4.8.

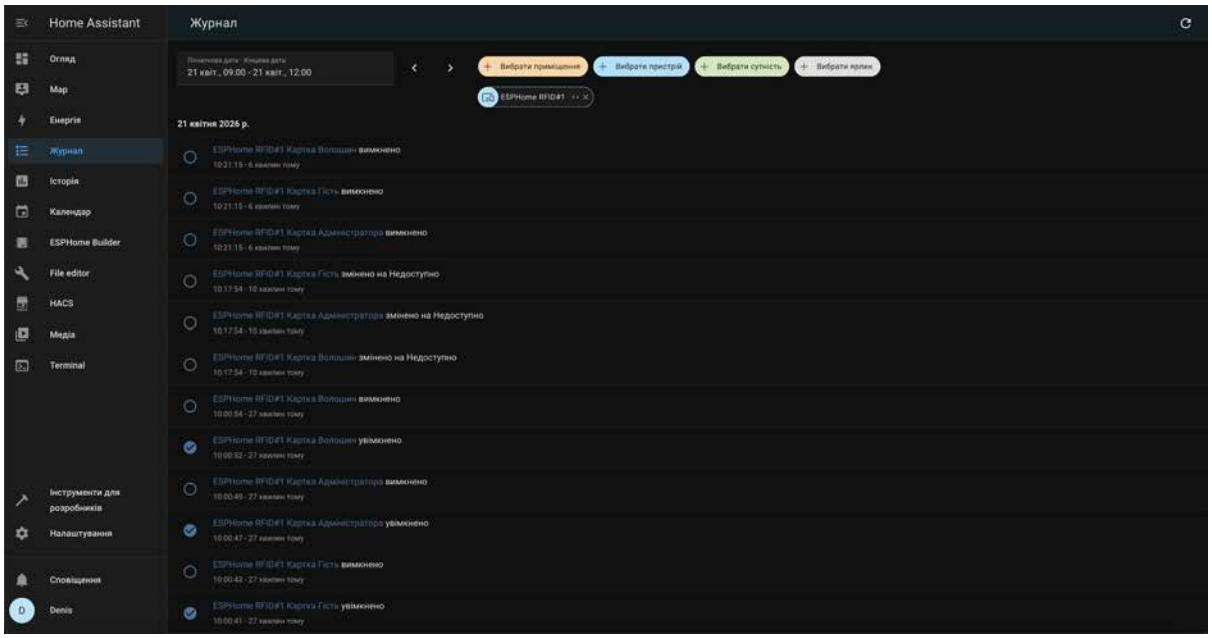

Рис. 4.8. Журнал подій підсистеми контролю відвідувачів

Журналювання подій є однією з найважливіших функцій системи, оскільки саме воно забезпечує накопичення даних про реальну роботу підсистеми доступу.

У межах даного проєкту журналювання потрібне для фіксації всіх спроб проходу, перегляду історії роботи системи, формування звітів, виявлення інцидентних ситуацій та контролю технічної активності вузлів.

У середовищі Home Assistant результати роботи системи можуть бути представлені через журнал подій, історію, інформаційні картки Dashboard, сторінки сутностей і пристроїв. Практична перевага такого підходу полягає в тому, що навіть без окремого спеціалізованого модуля звітності система вже на рівні Home Assistant надає достатньо інструментів для моніторингу й аналізу своєї роботи.

4.4. Реалізація інтерфейсу користувача на базі Home Assistant Dashboard

Сторінка автоматизацій, пов’язаних із контролем доступу наведена на рис. 4.9.

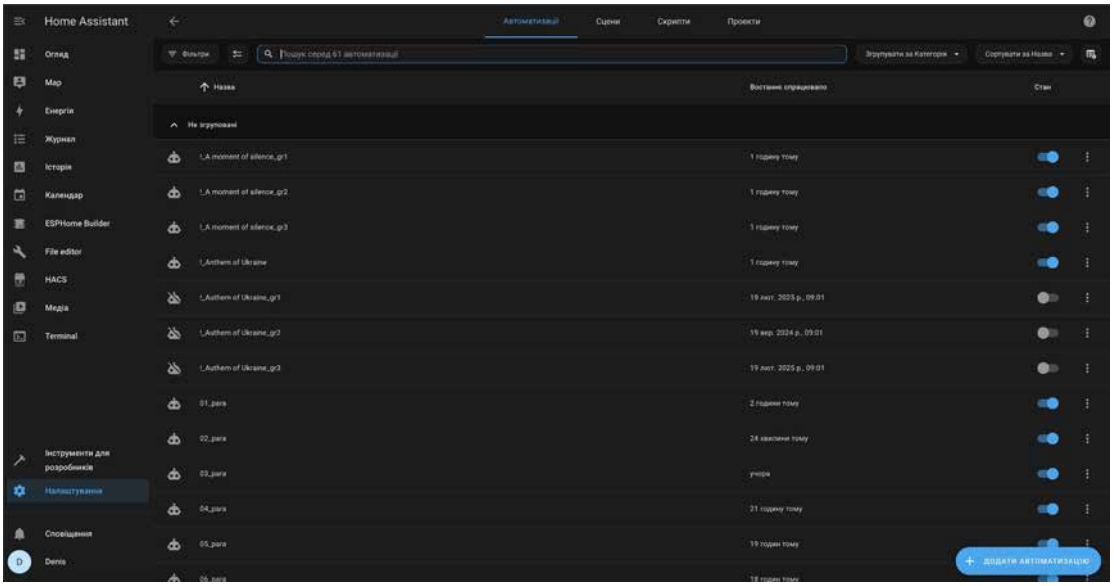

Рис. 4.9. Автоматизації підсистеми контролю відвідувачів

Інтерфейс користувача на базі Home Assistant Dashboard наведена на рис. 4.10.

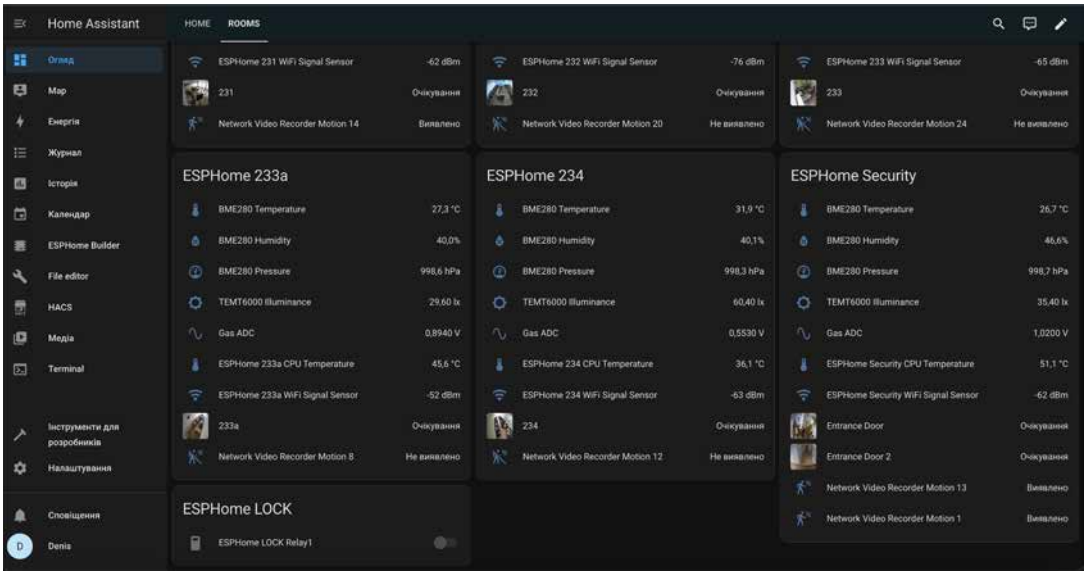

Рис. 4.10. Інтерфейс користувача підсистеми контролю відвідувачів

Інтерфейс користувача реалізується через Home Assistant Dashboard і пов’язані сторінки системи. Це дає змогу обійтися без створення окремого клієнтського застосунку та водночас забезпечити повноцінне відображення станів, подій і технічної інформації.

Через Dashboard адміністратор або охорона можуть переглядати статус вузла, останні події проходу, історію, журнал, автоматизації та сторінки пристроїв. Таким чином, інтерфейс виконує не лише інформаційну, а й організаційну функцію.

Перевага такого підходу полягає в тому, що система вже на етапі прототипу має зручне веб-середовище для моніторингу, а розробник може зосередитися на логіці доступу та інтеграції з обладнанням.

YAML-конфігурація вузла ESPHome RFID#1 наведена на рис. 4.11.

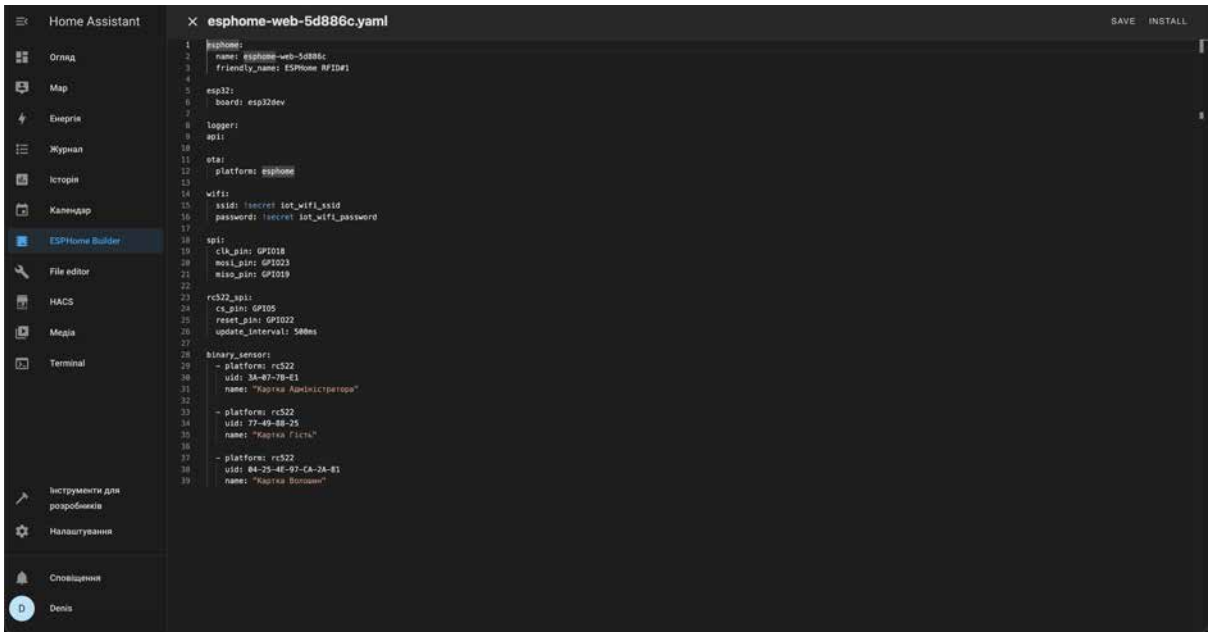

Рис. 4.11. Фрагмент YAML-конфігурації вузла ESPHome RFID#1

Практична цінність реалізованої системи суттєво зростає завдяки можливості продемонструвати її функціонування в реальному середовищі Home Assistant. Для

цього використовуються скріншоти головного екрана, ESPHome Builder, сторінки вузла ESPHome RFID#1, журналу подій, історії та автоматизацій.

Кожен із зазначених елементів підтверджує окремий аспект роботи системи: наявність серверного середовища, інтеграцію апаратного вузла, запуск логіки доступу та представлення інформації в інтерфейсі користувача.

Для демонстраційної реалізації використовується RC522, оскільки цей модуль є доступним, недорогим і достатнім для побудови працездатного макета. Він дозволяє перевірити логіку зчитування, передавання подій і базової обробки проходу.

Водночас для реального впровадження більш доцільним є PN532, що підтримує NFC і краще відповідає сучасним сценаріям цифрової ідентифікації. Саме тому в роботі розмежовується макетний і перспективний варіанти апаратної реалізації.

Після реалізації основних компонентів підсистеми контролю відвідувачів важливим етапом є перевірка її працездатності в типових умовах експлуатації. У межах даної бакалаврської роботи тестування має не лише технічне, а й демонстраційне значення, оскільки воно дозволяє підтвердити коректність інтеграції апаратної та програмної частин системи, а також узгодженість поведінки системи з поставленими функціональними вимогами.

Першим етапом тестування є перевірка базової доступності вузла ESPHome RFID#1 у середовищі Home Assistant. На цьому етапі необхідно переконатися, що вузол відображається у списку пристроїв, має активний стан з’єднання та коректно передає інформацію в систему. Така перевірка є основою для всіх подальших дій, оскільки без стабільного підключення апаратного вузла реалізація логіки контролю доступу не має практичного сенсу.

Другим етапом є тестування сценарію зчитування зареєстрованої мітки. У цьому випадку користувач підносить мітку до зчитувача, після чого система повинна:

- зафіксувати факт зчитування;
- передати UID у Home Assistant;
- встановити відповідність між міткою та користувачем;
- перевірити роль і правила доступу;
- сформувати рішення про дозвіл проходу;
- відобразити результат у журналі подій.

Успішне виконання цього сценарію підтверджує, що система реалізує свою основну функцію − контрольований доступ зареєстрованого користувача.

Третім етапом є перевірка сценарію відмови в доступі. Для цього використовується мітка, яка відсутня в системі, є неактивною або не має прав доступу до відповідної зони. У такому випадку система повинна:

- зафіксувати факт зчитування;
- виконати перевірку мітки;
- сформувати рішення про відмову;
- зафіксувати причину відмови;
- занести подію до журналу;
- за потреби створити повідомлення про інцидент.

Даний сценарій є не менш важливим, ніж сценарій успішного проходу, оскільки саме він демонструє здатність системи не лише дозволяти доступ, а й правильно реагувати на порушення правил проходу.

Четвертим етапом є перевірка відображення подій у середовищі Home Assistant. На цьому етапі тестується не апаратний, а інтерфейсний і моніторинговий рівень системи. Зокрема, потрібно переконатися, що події проходу з’являються в

журналі, відображаються в історії станів і можуть бути переглянуті через інформаційні панелі або сторінки пристроїв.

П’ятим етапом є перевірка автоматизацій, якщо вони використовуються в системі. Необхідно впевнитися, що після надходження події зчитування мітки запускається відповідна логіка, а результати її роботи узгоджуються з очікуваною поведінкою підсистеми.

Таким чином, тестування і верифікація системи в межах цієї роботи охоплюють як базові технічні перевірки підключення та зчитування, так і перевірку коректності робочих сценаріїв. Це дозволяє зробити висновок, що реалізована підсистема є працездатною, логічно узгодженою та придатною до подальшого розширення.

Після завершення первинного налаштування система переходить у режим повсякденної експлуатації. У цьому режимі її головне завдання полягає не в демонстрації окремих технічних можливостей, а в стабільному виконанні повторюваних операцій доступу й моніторингу. Для користувача взаємодія із системою повинна залишатися максимально простою: прикладання мітки до зчитувача, очікування результату перевірки та, за потреби, отримання доступу або відмови. Водночас для адміністратора та оператора охорони система повинна надавати достатньо детальну картину того, що відбувається всередині.

У реальній експлуатації важливо, щоб користувач не був змушений виконувати зайві дії. Саме тому в розроблюваній підсистемі основна складність перенесена із фронтальної взаємодії на внутрішню логіку обробки події. Користувач бачить лише кінцевий результат, тоді як перевірка ролі, пошук правил доступу, фіксація в журналі й оновлення інформації в середовищі Home Assistant відбуваються автоматично. Такий підхід відповідає загальним принципам проєктування інформаційних систем: складність повинна бути прихована від кінцевого користувача настільки, наскільки це можливо.

Інформаційні панелі в Home Assistant мають не лише декоративне, а й прикладне значення. Вони дозволяють звести на одному екрані ключові стани системи, останні події, індикатори активності вузлів, переходи до журналу, історії та сторінок пристроїв. Для системи контролю відвідувачів це особливо важливо, адже оператору потрібен не лише доступ до окремих сутностей, а й цілісне уявлення про стан підсистеми в поточний момент часу.

Під час формування таких панелей доцільно дотримуватися принципу ієрархічності. На верхньому рівні повинна відображатися узагальнена інформація: чи активний вузол, коли зафіксована остання подія, чи є інцидентні записи. На нижчому рівні можуть міститися докладніші елементи: журнали, історія, сторінки конфігурації, автоматизації та списки пристроїв. Така побудова інтерфейсу полегшує використання системи й робить її зрозумілішою навіть для користувача, який не брав безпосередньої участі у розробці.

Контроль коректності роботи підсистеми контролю відвідувачів є необхідною умовою підтвердження її практичної придатності. На відміну від звичайної демонстрації окремих функцій, методика перевірки результатів дозволяє системно оцінити, чи відповідає поведінка реалізованої системи поставленим вимогам і чи можна вважати її внутрішньо узгодженою.

У межах даної роботи перевірка коректності повинна здійснюватися на кількох рівнях.

Перший рівень − технічний контроль стану вузла. На цьому рівні перевіряється:

- чи відображається ESPHome RFID#1 у Home Assistant;
- чи є стабільним з’єднання вузла із серверною частиною;
- чи коректно працює ESPHome Builder;
- чи доступні пов’язані сутності пристрою.

Якщо на цьому етапі виникають помилки, подальша перевірка сценаріїв доступу не матиме достовірного результату.

Другий рівень − контроль обробки події зчитування. Для цього тестується, чи потрапляє зчитаний UID у цифрове середовище, чи змінюється стан відповідної сутності, чи з’являється подія в журналі та чи відображається зміна в історії. Це дозволяє підтвердити, що фізична подія зчитування справді проходить шлях до серверної логіки.

Третій рівень − контроль правильності логічного рішення. У межах цього рівня перевіряється, чи відповідає результат системи заданим умовам:

- зареєстрована мітка з дозволеним доступом повинна давати позитивний результат;
- невідома або неактивна мітка повинна викликати відмову;
- доступ у заборонений час або до недозволеної зони також має завершуватися відмовою.

Цей рівень є ключовим, оскільки саме він підтверджує, що система не просто реагує на зчитування, а коректно реалізує логіку правил доступу.

Четвертий рівень − контроль журналювання та представлення результатів. Тут перевіряється, чи відображаються події в журналі, чи фіксується результат проходу, чи можна простежити історію змін станів та чи доступна ця інформація через інтерфейс Home Assistant. Якщо події обробляються, але не відображаються в журналі або історії, система втрачає частину своєї інформаційної цінності.

П’ятий рівень − контроль повторюваності результатів. Для достовірності необхідно перевірити, чи дає система однаковий результат за однакових умов у кількох повторних тестах. Такий підхід дозволяє уникнути ситуації, коли працездатність системи підтверджується лише одноразовим успішним запуском.

Методика перевірки результатів у цій роботі може бути подана у вигляді послідовності контрольних сценаріїв:

1) перевірка підключення вузла;
2) перевірка зчитування мітки;
3) перевірка позитивного сценарію доступу;
4) перевірка негативного сценарію доступу;
5) перевірка журналу подій;
6) перевірка історії станів;
7) перевірка автоматизацій;
8) порівняння фактичного результату з очікуваним.

Такий підхід дає змогу оцінити систему комплексно: від фізичного рівня до інтерфейсного відображення результатів.

Отже, контроль коректності роботи системи та методика перевірки результатів дозволяють підтвердити, що реалізована підсистема контролю відвідувачів працює послідовно, логічно правильно та відповідає вимогам, сформованим на етапі проєктування. Це є важливою підставою для подальшого використання системи як демонстраційного прототипу та основи для майбутнього впровадження.

Навіть у випадку, коли система створюється як прототип, доцільно одразу окреслити базові рекомендації щодо її подальшого впровадження та супроводу. По-перше, перед початком практичного використання потрібно впорядкувати перелік ролей, зон і точок проходу, щоб логіка доступу відображала реальну організаційну структуру закладу. По-друге, необхідно визначити відповідальних осіб за адміністрування міток, підтримку конфігурацій та контроль журналів подій. По-третє, важливо ще на старті домовитися про правила перевидачі, блокування та архівування міток, інакше навіть добре побудована система з часом втратить керованість.

Не менш важливою є організація технічного супроводу. Серверна частина на базі Home Assistant повинна регулярно резервуватися, а конфігурації вузлів

ESPHome − зберігатися в окремому контрольованому сховищі. Для системи доступу це критично, оскільки втрата конфігурації або пошкодження середовища може призвести не лише до технічних незручностей, а й до порушення організаційних процесів. Саме тому в реальному впровадженні доцільно запроваджувати хоча б базовий регламент резервного копіювання та відновлення.

Окрему увагу слід приділити інформаційній культурі користувачів. Практика показує, що ефективність системи доступу залежить не лише від якості обладнання та програмного забезпечення, а й від того, наскільки зрозумілими для користувачів є правила роботи з мітками та обмеження доступу. Тому поряд із технічним впровадженням доцільно готувати короткі інструкції для персоналу, адміністратора та охорони, де буде описано основні сценарії: видача мітки, повідомлення про втрату, блокування, отримання тимчасового доступу та порядок реагування на інцидентні записи.

Щоб оцінити працездатність реалізованої підсистеми, недостатньо лише зафіксувати факт появи вузла в Home Assistant. Необхідно послідовно перевірити весь ланцюг проходження події: від зчитування мітки до відображення результату в журналі або на інформаційній панелі. Для цього доцільно використовувати серію контрольних сценаріїв, кожен з яких перевіряє окрему частину системи.

До таких контрольних сценаріїв можна віднести: перевірку підключення вузла до серверної частини; перевірку доступності сторінки ESPHome Builder; перевірку відкриття сторінки пристрою й наявності пов'язаних сутностей; тестове зчитування мітки; фіксацію події в журналі; аналіз зміни станів в історії; перевірку реакції автоматизацій. Якщо всі ці кроки виконуються коректно, можна зробити висновок, що система реалізована цілісно, а не складається з окремих слабо пов'язаних фрагментів.

Після реалізації основних компонентів системи важливим етапом є перевірка працездатності та узгодженості її поведінки в типових сценаріях. Для бакалаврської роботи тестування має не лише технічний, а й демонстраційний характер.

Найпростішим тестовим сценарієм є перевірка доступу для зареєстрованої мітки. У цьому випадку користувач підносить мітку до зчитувача, система отримує UID, знаходить відповідний запис, виконує перевірку ролі й правил доступу та фіксує подію як успішну.

Другий тестовий сценарій полягає у використанні мітки, яка відсутня в системі або має неактивний статус. У такому випадку очікується негативний результат перевірки, запис події як відмови та, за наявності відповідної логіки, формування інцидентного повідомлення.

Окремо слід перевірити інтерфейсну частину системи. Для цього аналізуються сторінки Dashboard, журнал подій, історія та ESPHome Builder. Якщо система коректно показує стан вузла, появу події та результат проходу, можна вважати, що інтерфейсний рівень реалізований узгоджено з логікою системи.

Висновки до розділу 4

У четвертому розділі було розглянуто практичну реалізацію підсистеми контролю відвідувачів для Smart Campus на базі Home Assistant. Установлено, що використання цієї платформи як центрального програмного середовища є доцільним, оскільки вона поєднує в собі засоби інтеграції пристроїв, інтерфейс користувача, механізми автоматизації, журналювання подій та моніторинг станів у межах єдиної системи.

У ході розгляду практичної частини описано порядок розгортання та початкового налаштування серверного середовища, а також показано роль ESPHome як проміжного рівня між мікроконтролерним вузлом і Home Assistant. Визначено, що вузол ESPHome RFID#1 виконує функцію реальної апаратно-

програмної точки доступу, через яку забезпечується зв’язок між фізичним зчитуванням мітки та обробкою події в цифровому середовищі Smart Campus.

Окрему увагу приділено реалізації логіки перевірки доступу. Показано, що система виконує послідовну обробку події: отримує ідентифікатор мітки, встановлює її належність конкретному користувачу, визначає роль, перевіряє правила доступу, формує рішення про дозвіл або відмову та фіксує результат у журналі. Такий підхід дозволяє реалізувати не лише технічний механізм проходу, а повноцінну інформаційну логіку контролю доступу.

У розділі також обґрунтовано підхід до журналювання подій та відображення результатів у Home Assistant. Визначено, що журнал подій, історія змін станів, сторінки пристроїв, ESPHome Builder та інформаційні панелі Dashboard формують достатнє середовище для моніторингу, аналізу та супроводу системи без необхідності розроблення окремого спеціалізованого клієнтського застосунку.

Крім того, було показано, що для демонстраційної реалізації підсистеми доцільно використовувати RC522 як доступний і технічно достатній модуль для побудови працездатного макета. Водночас для реального практичного впровадження більш перспективним рішенням є PN532, який підтримує NFC та краще відповідає сучасним вимогам до цифрової ідентифікації користувачів.

Отже, у четвертому розділі підтверджено, що підсистема контролю відвідувачів може бути практично реалізована на базі Home Assistant, ESPHome та ESP32, інтегрована в середовище Smart Campus і використана як основа для подальшого розвитку повноцінної системи контролю доступу в закладі вищої освіти.

РОЗДІЛ 5 АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ ТА МОЖЛИВОСТІ ЇЇ ПОДАЛЬШОГО РОЗВИТКУ

5.1. Аналіз функціональних можливостей реалізованої системи

Після реалізації основних елементів підсистеми контролю відвідувачів доцільно оцінити її функціональні можливості, практичну придатність та відповідність вимогам, сформованим на етапі проєктування. У межах даної бакалаврської роботи реалізована система має прототипний характер, однак її склад і поведінка дозволяють перевірити працездатність обраної архітектури та логіки функціонування.

Насамперед слід зазначити, що система забезпечує ідентифікацію користувача через мітку доступу. Це є базовою функцією будь-якої підсистеми контролю проходу. У контексті Smart Campus така функція набуває більшого значення, оскільки вона не обмежується лише розпізнаванням UID мітки, а пов’язується з даними про користувача, його роль, правила доступу та точку проходу.

Другою важливою функціональною можливістю є інтеграція фізичної події з цифровим середовищем Home Assistant. Реалізована система демонструє, що спроба проходу не залишається локальною подією на рівні апаратного вузла, а стає елементом загального інформаційного простору Smart Campus.

Третьою функціональною можливістю є перевірка правил доступу. У системі передбачається, що рішення про прохід залежить не лише від факту наявності мітки, а від сукупності умов: належності мітки до користувача, ролі користувача, відповідності точки проходу допустимій зоні та можливих часових обмежень.

Четвертою суттєвою функціональною можливістю є журналювання подій. Події проходу, відмови, зміни станів вузла та інші дії можуть бути переглянуті в

межах Home Assistant. Це дає змогу не лише контролювати поточну ситуацію, а й аналізувати історію роботи системи.

П’ятою важливою функцією є наявність користувацького інтерфейсу без розробки окремого спеціалізованого додатка. Для бакалаврського проєкту це є істотною перевагою, оскільки значно спрощує реалізацію й одночасно підвищує практичну цінність рішення.

Разом із тим реалізована система має і певні обмеження. Зокрема, у поточному вигляді вона не містить розподіленого адміністрування для великої кількості будівель, централізованого багаторівневого механізму керування доступом для численних підрозділів, окремого резервного контуру відмовостійкості та глибокої інтеграції з іншими системами безпеки.

Таким чином, аналіз функціональних можливостей показує, що система вже на рівні прототипу забезпечує основні процеси ідентифікації, перевірки доступу, журналювання, моніторингу та базового адміністрування.

Сценарій успішного доступу

У базовому сценарії користувач підносить мітку до зчитувача. Після цього ідентифікатор мітки надходить до вузла ESP32, передається в Home Assistant і далі використовується логікою системи для перевірки прав доступу. Якщо мітка є зареєстрованою, а користувач має право доступу до відповідної зони в поточний момент часу, система фіксує подію як успішну.

З погляду аналізу цей сценарій демонструє коректність технічної взаємодії апаратного та програмного рівнів, наявність логічного зв’язку між міткою та користувачем, працездатність механізму правил доступу та здатність системи фіксувати позитивний результат у журналі.

Сценарій відмови в доступі

Іншим важливим сценарієм є відмова в доступі. Такий результат може бути зумовлений різними причинами: мітка відсутня в системі, мітка деактивована,

користувач не має права доступу до заданої зони, правило доступу тимчасово неактивне, поточний час не входить у дозволений інтервал.

У цьому сценарії цінність системи проявляється не лише у факті блокування проходу, а й у фіксації причини відмови. Це дозволяє відрізняти звичайну помилку від потенційного інциденту.

Сценарій моніторингу охороною

Для служби охорони важливим є не процес керування мітками як такий, а можливість швидко переглядати події, які вже відбулися, а також реагувати на нештатні ситуації. У цьому сценарії користувач відкриває Home Assistant, переходить до журналу подій або історії та переглядає зміни станів, пов’язані з точкою доступу.

Сценарій адміністрування

Адміністратор системи може аналізувати стан вузлів, редагувати конфігурації, працювати з автоматизаціями, відстежувати технічні журнали. У такому сценарії важливо, що вся система зосереджена в одному середовищі й не потребує використання великої кількості розрізнених інструментів.

Отже, аналіз сценаріїв роботи системи підтверджує, що підсистема контролю відвідувачів є придатною як для демонстрації базової логіки доступу, так і для вирішення реальних задач моніторингу та адміністрування.

Приклад вибірки подій проходу за певний період наведено на рис. 5.3.

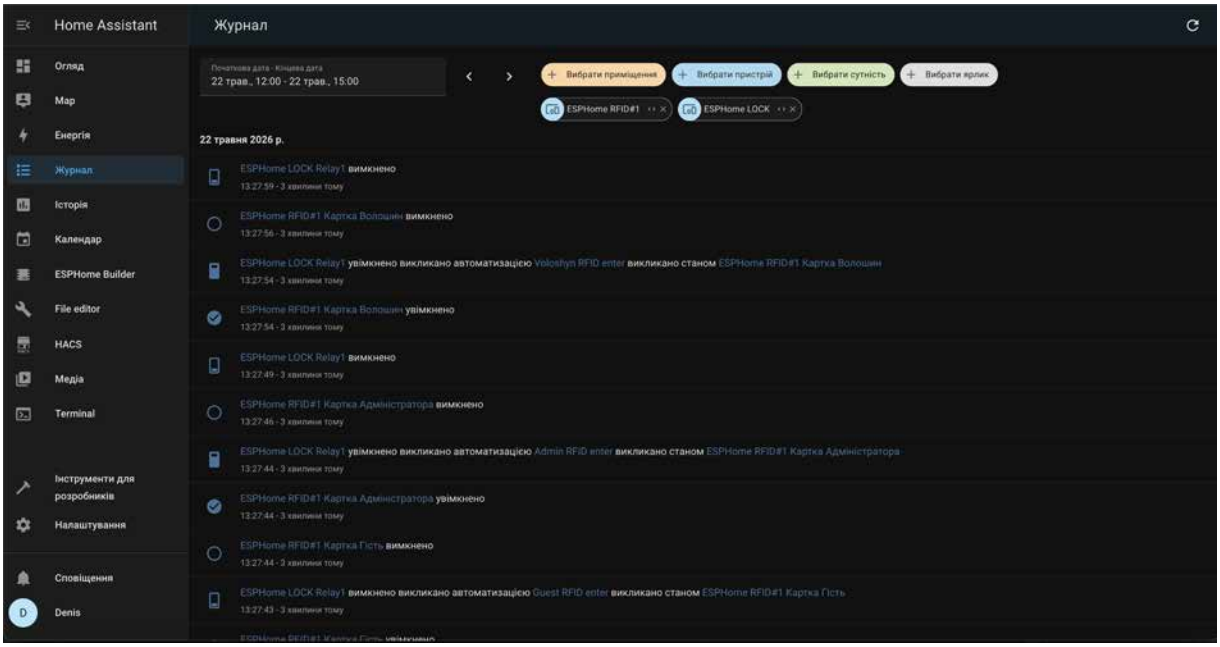

Рис. 5.3. Приклад вибірки подій проходу за певний період

Приклад звітно-статистичної інформації системи на рис. 5.4.

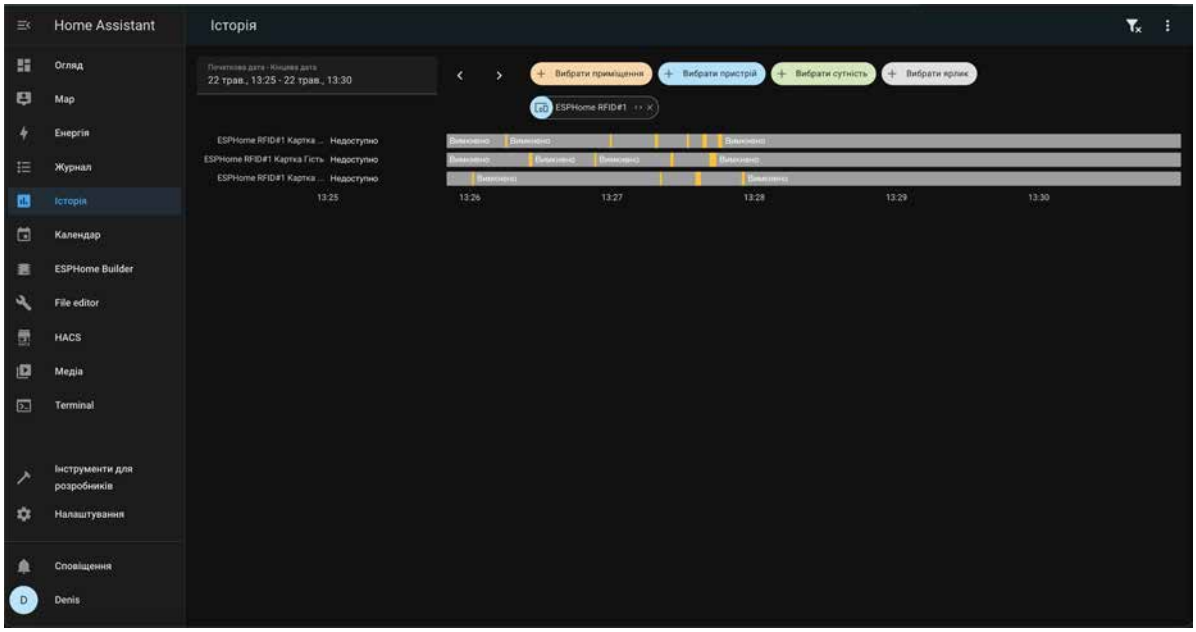

Рис. 5.4. Приклад звітно-статистичної інформації системи

Однією з важливих функцій підсистеми контролю відвідувачів є накопичення подій та можливість подальшого формування звітної й статистичної інформації. Це

суттєво відрізняє інформаційну систему від простого механізму локального доступу, який лише відкриває або блокує двері.

У межах даного проєкту базою для формування звітної інформації є журнал подій Home Assistant, історія змін станів, дані вузла ESPHome RFID#1 та таблиці бази даних PostgreSQL.

Першим рівнем звітності є оперативний перегляд подій. Другим рівнем є структурована звітність на основі БД. Якщо події зберігаються у таблицях системи, то можна будувати більш точні запити: звіт про всі проходи за день, звіт про кількість успішних і неуспішних спроб, статистику по окремих точках проходу, статистику по користувачах або ролях, перелік інцидентних подій за заданий період.

Третім рівнем є статистичне узагальнення. На основі накопичених подій можна формувати не лише списки, а й агреговану інформацію: кількість проходів по днях, співвідношення дозволених і заборонених спроб, частоту використання конкретних точок доступу, динаміку подій у часі.

Таким чином, система контролю відвідувачів на базі Home Assistant дозволяє не лише реалізовувати контроль проходу, а й накопичувати події, придатні для побудови журналів, звітів і статистики, що підвищує її значення в межах Smart Campus.

5.2. Практична придатність системи для використання в університетському середовищі

Одним із важливих завдань заключного розділу є оцінка того, наскільки розроблена система може бути використана не лише як навчальний макет, а як практична основа для подальшого впровадження в університетському середовищі.

Практична придатність системи проявляється в кількох аспектах. Технічна придатність полягає в тому, що система побудована на доступних, реальних і

відносно простих для впровадження компонентах: Home Assistant, ESPHome, ESP32, RFID/NFC-зчитувач, PostgreSQL. Організаційно система є придатною тому, що в її межах вже закладено логіку ролей, зон і правил доступу. Експлуатаційна придатність забезпечується використанням Home Assistant як інтерфейсу користувача.

Таким чином, навіть у поточному вигляді система має практичну придатність як мінімум для пілотного розгортання, демонстраційного використання, внутрішнього тестового контуру та навчального застосування в лабораторному середовищі.

На основі проведеного аналізу можна виділити основні переваги розробленої системи: інтегрованість, використання Home Assistant як єдиного середовища для серверної логіки, інтерфейсу користувача, автоматизацій, журналів і моніторингу, модульна структура, реальна фізична реалізація вузла ESPHome RFID#1, можливість журналювання та формування звітної інформації, а також можливість розвитку в напрямі NFC.

Попри очевидні переваги, реалізоване рішення має і певні обмеження, які потрібно враховувати чесно й обґрунтовано. По-перше, поточна реалізація є прототипною, а не повномасштабною університетською системою. По-друге, використання RC522 є технічно достатнім для демонстрації, але не є найкращим варіантом для повноцінного перспективного впровадження. По-третє, система в межах цієї роботи не має окремого промислового механізму резервування, багатовузлової централізації та повного контуру відмовостійкості. По-четверте, поточна реалізація не включає глибоку інтеграцію з усіма можливими підсистемами безпеки.

Подальший розвиток системи може здійснюватися за кількома основними напрямами: перехід до PN532 і NFC, розширення кількості точок доступу, розвиток

аналітичного модуля, поглиблення інтеграції зі Smart Campus, розвиток інтерфейсу адміністрування.

Таким чином, система має достатній потенціал для подальшого переходу від прототипного рівня до практично впроваджуваної підсистеми.

Для повнішого аналізу отриманих результатів доцільно оцінити не лише функціональні можливості розробленої системи, а й її ефективність з практичного погляду. У межах бакалаврської роботи підсистема реалізована у вигляді прототипу, уже на цьому етапі можна визначити, який прикладний ефект дає її впровадження в середовище Smart Campus.

Першим аспектом ефективності є підвищення керованості доступу до приміщень. У разі використання розробленої системи рішення про допуск приймається не вручну і не на основі розрізнених локальних механізмів, а в межах єдиної логіки, що враховує мітку, роль користувача, точку проходу та правила доступу. Це дає змогу зменшити кількість помилок, пов’язаних із людським фактором, і забезпечити більш впорядковану організацію проходу.

Другим аспектом є скорочення часу на перевірку доступу. У традиційних або частково ручних схемах доступу перевірка прав користувача може вимагати додаткових дій з боку охорони або адміністратора. У запропонованій системі основна перевірка виконується автоматично, а отже процес проходу стає швидшим і зручнішим для кінцевого користувача.

Третім аспектом ефективності є підвищення інформаційної прозорості. Система не лише дозволяє або забороняє прохід, а й фіксує всі події, що відбуваються. Це дає можливість:

- переглядати історію проходів;
- виявляти неуспішні спроби доступу;
- оцінювати навантаження на окремі точки проходу;
- аналізувати поведінку системи в часі.

Така прозорість є важливою як для технічного супроводу, так і для організаційного контролю.

Четвертим аспектом є гнучкість адміністрування. Завдяки використанню ролей, зон і правил доступу система може бути адаптована до змін у структурі університету без повної перебудови архітектури. Наприклад, можна змінити правила доступу для певної категорії користувачів або додати нову точку проходу без потреби змінювати всю логіку системи.

П’ятим аспектом є зменшення залежності від ізольованих технічних рішень. Оскільки розроблена підсистема інтегрується в Home Assistant, вона не функціонує як окремий замкнений механізм, а стає частиною більш широкого цифрового середовища Smart Campus. Це створює умови для подальшого поєднання із системами сповіщення, аналітики, моніторингу та іншими підсистемами кампусу.

Якщо оцінювати практичний ефект від упровадження системи в цілому, то його можна узагальнити такими положеннями:

- підвищення рівня організованості доступу;
- покращення контролю за подіями проходу;
- створення єдиного інформаційного середовища для моніторингу;
- зростання зручності користування системою;
- формування основи для подальшого масштабування.

Отже, оцінювання ефективності показує, що розроблена система має не лише демонстраційну, а й реальну прикладну цінність. Уже на рівні прототипу вона створює відчутний практичний ефект у вигляді автоматизації проходу, впорядкування доступу, централізації подій та підвищення керованості цифрового середовища Smart Campus.

Ефект від впровадження системи контролю відвідувачів не обмежується лише автоматизацією проходу. На організаційному рівні така система дозволяє впорядкувати доступ до зон університету, зменшити залежність від паперових або

усних механізмів пропуску, а також створити прозорий механізм фіксації подій. Для адміністрації це означає підвищення керованості простором, а для охорони — появу додаткового інструменту спостереження та аналізу.

З безпекового погляду особливу цінність має не лише сама можливість відмови в доступі, а й факт накопичення історії подій. У разі спірної або підозрілої ситуації система дозволяє встановити, коли саме була здійснена спроба проходу, на якій точці, з використанням якої мітки та з яким результатом. Така інформація має вагоме значення для подальшого розгляду інцидентів і може використовуватися як допоміжний інструмент внутрішнього аудиту.

Однією з сильних сторін запропонованого підходу є те, що він не прив'язаний лише до одного зчитувача чи одного приміщення. Структура системи допускає розширення на кілька точок проходу, кілька зон і навіть кілька корпусів. Для цього не потрібно змінювати базову ідеологію підсистеми достатньо додавати нові записи до довідників зон, точок доступу, правил і, за потреби, підключати додаткові вузли ESPHome.

У межах університетського середовища це особливо важливо, оскільки реальний кампус є розподіленим простором. Якщо підсистема успішно працює на рівні одного вузла, наступним логічним кроком може стати поетапне підключення нових проходів: спершу в межах одного корпусу, далі між корпусами, а потім на рівні комплексної інфраструктури. Така поступова стратегія впровадження виглядає набагато реалістичнішою, ніж спроба одномоментно автоматизувати весь кампус.

Проведене дослідження дозволило комплексно розглянути задачу побудови інформаційної системи контролю відвідувачів для Smart Campus на основі Home Assistant від аналізу предметної області до практичної реалізації прототипу та оцінки його прикладних можливостей.

У ході роботи було встановлено, що система контролю відвідувачів у сучасному університетському середовищі не повинна розглядатися лише як технічний механізм відкривання або блокування проходу. Натомість вона має бути інтегрованою інформаційною підсистемою, яка поєднує фізичну ідентифікацію користувача, логіку перевірки прав доступу, журналювання подій, інтерфейсне представлення та можливість подальшого аналізу накопичених даних.

Результати дослідження показали, що для реалізації такої підсистеми доцільно використовувати платформу Home Assistant. Її перевага полягає в тому, що вона дозволяє в одному середовищі поєднати:

- інтеграцію апаратних вузлів;
- серверну логіку обробки подій;
- засоби автоматизації;
- інтерфейс користувача;
- журнал подій та історію станів.

Саме така цілісність робить обраний підхід переконливим не лише як лабораторний приклад, а і як основу для практичного розвитку системи.

Важливим результатом дослідження стало проєктування логічної та фізичної моделей бази даних. Це дозволило перейти від загальної ідеї до формалізованої структури інформаційної бази, яка підтримує ролі користувачів, мітки доступу, зони, точки проходу, правила доступу, події та інцидентні повідомлення. Наявність такої моделі підтверджує, що система розглядається як повноцінне інформаційне рішення, а не лише як набір окремих апаратних дій.

Окремого значення набули результати практичної частини роботи. Реалізація підсистеми на базі ESP32, ESPHome та Home Assistant довела, що запропонований підхід може бути втілений не лише теоретично, а й у реальному функціонуючому макеті. Використання вузла ESPHome RFID#1 як реальної точки доступу показало

можливість побудови зв’язку між фізичним зчитуванням мітки та цифровою обробкою події в середовищі Smart Campus.

Крім того, у роботі було підтверджено, що система є придатною до подальшого розширення. Поєднання модульної архітектури, реляційної бази даних, Home Assistant як центральної платформи та можливості переходу від RC522 до PN532 створює основу для майбутнього масштабування системи на кілька точок проходу, зон доступу та, за потреби, кілька корпусів університету.

Узагальнюючи результати дослідження, можна стверджувати, що в межах бакалаврської роботи вдалося досягти головного: запропонувати, обґрунтувати та практично продемонструвати підхід до побудови підсистеми контролю відвідувачів, яка:

- логічно вписується у концепцію Smart Campus;
- базується на доступних програмних та апаратних компонентах;
- забезпечує інтегроване середовище моніторингу і керування;
- має прикладну цінність для подальшого розвитку.

Отже, результати виконаного дослідження свідчать про те, що побудова інформаційної системи контролю відвідувачів для Smart Campus на основі Home Assistant є технічно можливою, методично обґрунтованою та практично перспективною. Саме це і становить основний підсумок проведеної роботи.

Щоб більш обґрунтовано оцінити отриманий результат, доцільно порівняти запропонований підхід із двома типовими альтернативами: автономною локальною системою доступу без централізованої інтеграції та комерційною спеціалізованою системою контролю доступу. Автономне локальне рішення є простішим у початковому розгортанні, однак має обмежені можливості аналітики, слабку інтеграцію з цифровим середовищем і часто погано масштабується. Комерційні системи, навпаки, можуть мати багатший набір функцій, але потребують вищих витрат, прив'язують користувача до конкретного вендора та не завжди

забезпечують ту гнучкість інтеграції, яка потрібна саме в університетському експериментальному середовищі.

Розроблений у цій роботі підхід займає проміжну позицію. Він є суттєво гнучкішим і більш інтегрованим, ніж простий автономний контролер, але при цьому залишається доступнішим і зрозумілішим для навчального або пілотного використання, ніж повноцінна комерційна платформа. Саме це можна вважати однією з головних переваг запропонованої системи: вона дозволяє розвивати Smart Campus етапно, не відмовляючись від власного контролю над архітектурою, конфігураціями та накопиченими даними.

Ще один аспект порівняння стосується інтерфейсного рівня. У класичних автономних рішеннях користувач часто працює або з мінімальним локальним дисплеєм, або взагалі не має зручного засобу перегляду історії подій. У комерційних рішеннях інтерфейс, як правило, є повноцінним, але закритим і жорстко заданим виробником. У випадку Home Assistant інтерфейс формується гнучко: його можна адаптувати під вимоги адміністратора, охорони або окремого тестового сценарію. Для університетського середовища, де цифрова інфраструктура може змінюватися та доповнюватися, це є вагомою практичною перевагою.

Проведене дослідження показало, що побудова підсистеми контролю відвідувачів на базі Home Assistant є технічно можливою й методично виправданою. Важливою перевагою такого підходу є поєднання кількох рівнів у межах одного середовища: інтеграції апаратних вузлів, серверної логіки, візуального інтерфейсу, журналювання та базового аналітичного представлення даних. Саме ця цілісність робить рішення переконливим не лише як лабораторний макет, а як основу для подальшого розвитку.

Узагальнюючи результати, можна зробити висновок, що в роботі вдалося досягти головного: запропонувати та описати систему, яка логічно вписується у

концепцію Smart Campus, використовує доступні технології, має реальну демонстраційну реалізацію й допускає подальше масштабування. Саме поєднання доступності, модульності й інтегрованості є головною цінністю розробленого підходу.

Для повнішого аналізу результатів доцільно оцінити не лише функціональні можливості системи, а й очікуваний практичний ефект від її впровадження. Навіть якщо у межах бакалаврської роботи система представлена у вигляді прототипу, уже на цьому етапі можна визначити напрями, у яких її використання створює додану цінність для університетського середовища.

Перший ефект пов’язаний із підвищенням керованості доступу. Система дозволяє не покладатися лише на локальні рішення, а централізовано фіксувати, хто, коли та де намагався пройти.

Другий ефект полягає в покращенні інформаційної прозорості. Якщо події проходу накопичуються в єдиному контурі, то адміністрація або охорона можуть отримувати не лише окремі сигнали, а й узагальнену картину активності.

Третій ефект стосується гнучкості адміністрування. Використання ролей, зон і правил доступу робить систему придатною до адаптації без кардинальної зміни всієї архітектури.

Отже, навіть на рівні прототипу підсистема контролю відвідувачів має практичний ефект: підвищує структурованість доступу, покращує інформаційний контроль і створює основу для подальшого розвитку університетського цифрового середовища.

Висновки до розділу 5

У п’ятому розділі було виконано аналіз результатів роботи реалізованої системи контролю відвідувачів для Smart Campus на основі Home Assistant.

Показано, що система забезпечує базові функції ідентифікації, перевірки доступу, журналювання подій, інтерфейсного відображення та моніторингу.

Було проаналізовано типові сценарії роботи системи, а також можливості формування журналів, звітної й статистичної інформації. Встановлено, що запропоноване рішення має практичну придатність для використання в умовах університетського середовища як пілотна або демонстраційна підсистема.

Окремо визначено переваги, обмеження та перспективні напрями розвитку системи. Установлено, що розроблена архітектура є достатньо гнучкою та масштабованою, а подальше вдосконалення може бути пов’язане з переходом до PN532, збільшенням кількості точок доступу, розвитком аналітики та глибшою інтеграцією в Smart Campus.

ВИСНОВКИ

У бакалаврській роботі було вирішено актуальне завдання розробки інформаційної системи контролю відвідувачів для Smart Campus на основі Home Assistant.

Проаналізовано предметну область, обґрунтовано вибір RFID/NFC-підходу, побудовано UML-моделі, спроєктовано логічну та фізичну моделі бази даних, а також визначено програмно-апаратну структуру системи.

Обґрунтовано використання PostgreSQL як СУБД, Home Assistant як основної платформи реалізації та ESPHome RFID#1 як апаратно-програмної основи точки доступу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Національний університет біоресурсів і природокористування України. Положення про бакалаврську кваліфікаційну роботу у Національному університеті біоресурсів і природокористування України. Київ, 2021. 49 с.

2. Home Assistant. Documentation. URL: https://www.home-assistant.io/docs/ (дата звернення: 23.04.2026).

3. Home Assistant. Installation. URL: https://www.home-assistant.io/installation/ (дата звернення: 23.04.2026).

4. ESPHome. ESPHome Documentation. URL: https://esphome.io/components/ (дата звернення: 23.04.2026).

5. ESPHome. Getting Started with ESPHome and Home Assistant. URL: https://esphome.io/guides/getting_started_hassio/ (дата звернення: 23.04.2026).

6. PostgreSQL Global Development Group. PostgreSQL: Documentation. URL: https://www.postgresql.org/docs/ (дата звернення: 23.04.2026).

7. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. Boston: Addison-Wesley, 2004. 208 p.

8. Elmasri R., Navathe S. B. Fundamentals of Database Systems. 7th ed. Boston: Pearson, 2016. 1272 p.

9. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner’s Approach. 8th ed. New York: McGraw-Hill Education, 2015. 976 p.

10. Sommerville I. Software Engineering. 10th ed. Boston: Pearson, 2015. 816 p.

ДОДАТОК

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

Я, Медяник Денис Андрійович, здобувач НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота на тему «ІНФОРМАЦІЙНА СИСТЕМА КОНТРОЛЮ ВІДВІДУВАЧІВ ДЛЯ SMART CAMPUS НА ОСНОВІ HOME ASSISTANT» є результатом моєї особистої праці.

2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.

3. Щодо використання інструментів ШІ зазначаю, що:

[ ] інструменти генеративного ШІ не використовувалися.

[X] інструменти ШІ (ChatGPT, DeepL, були використані для:

[X] перекладу іншомовних джерел;

[X] стилістичного редагування та перевірки граматики;

[ ] допомоги у написанні/відлагодженні програмного коду;

[ ] інше.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«___»_____ 2026 р. _____ **Денис МЕДЯНИК**