НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ПОГОДЖЕНО	**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**
Декан факультету	**Завідувач кафедри комп'ютерних наук**
_____ **Ігор БОЛБОТ**	_____ **Белла ГОЛУБ**
(підпис)	(підпис)
«___» _____ 2026 р.	«___» _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Програмне забезпечення для системи керування доступом до будівель за допомогою цифрових ключів»

Спеціальність «121 Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

Гарант освітньої програми к.т.н., доцент	_____ (підпис)	**Ганна ВАЙГАНГ**
Керівник бакалаврської кваліфікаційної роботи к.т.н., с.н.с.	_____ (підпис)	**Віктор ПАНКРАТЬЄВ**
Виконав	_____ (підпис)	**Владислав СЕРДЮК**

Київ-2026

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук

к.т.н., доцент _____ Белла ГОЛУБ
(підпис)

«___» _____ 2025 р.

ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ

Сердюку Владиславу Сергійовичу
(прізвище, ім'я, по батькові)

Спеціальність «121 Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення»

Тема бакалаврської кваліфікаційної роботи

«Програмне забезпечення для системи керування доступом до будівель за допомогою цифрових ключів»

затверджена наказом від «19» грудня 2025 р. №3204 «С»
Термін подання завершеної роботи на кафедру «22 травня 2026 р.
Вихідні дані до бакалаврської кваліфікаційної роботи (проєкту):
_____
Перелік питань, що підлягають дослідженню:
1. Системи керування доступом до будівель
2. Проєктування інформаційного забезпечення
3. Розробка програмного забезпечення
4. Тестування ефективності системи
Перелік графічних документів (за необхідності).
Дата видачі завдання «19» грудня 2025 р.

Керівник бакалаврської кваліфікаційної роботи _____ (підпис) **Віктор ПАНКРАТЬЄВ**

Завдання взяв до виконання _____ (підпис) **Владислав СЕРДЮК**

ЗМІСТ

ВСТУП 5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СИСТЕМИ КЕРУВАННЯ ДОСТУПОМ ДО БУДІВЕЛЬ З ВИКОРИСТАННЯМ ЦИФРОВИХ КЛЮЧІВ 7
1.1 Опис предметної області систем керування доступом до будівель 7
1.2 Теоретико-методологічні засади використання цифрових ключів у системах доступу 9
1.3 Аналіз існуючих рішень й інформаційних систем 11
1.4 Моделювання предметної області 16
1.5 Аналіз вимог інформаційної системи 19
1.6 Постановка завдання 22
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 24
2.1 Логічна модель даних системи керування доступом до будівель з використанням цифрових ключів 24
2.2 Діаграма класів і кооперації системи керування доступом 27
2.3 Діаграма компонентів програмного забезпечення 30
2.4 Діаграма пакетів системи керування доступом 33
2.5 Висновки до другого розділу 34
3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 36
3.1 Обґрунтування вибору технологічних засобів реалізації 36
3.2 Розробка фізичної моделі бази даних 38
3.3 Алгоритмізація роботи програмних модулів системи 42
3.4 Реалізація основних програмних модулів 47
3.5 Опис інтерфейсу користувача 49
3.6 Висновки до третього розділу 51
4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ 53

4.1 План тестування програмних модулів та методика оцінювання результатів 53
4.2 Тестування системи керування доступом до будівель з використанням цифрових ключів 55
4.4 Висновки до четвертого розділу 63
ВИСНОВКИ 65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 67
ДОДАТОК А 70
ДОДАТОК Б 72
ДОДАТОК В 74
ДОДАТОК Г 75

ВСТУП

Актуальність теми зумовлена зростанням потреби у надійних, зручних та масштабованих засобах контролю доступу до будівель. Традиційні механічні ключі та пластикові картки поступово втрачають універсальність, оскільки їх складно оперативно відкликати, контролювати, передавати тимчасовим користувачам і пов’язувати з журналом подій. У сучасних офісних, навчальних, житлових і адміністративних будівлях важливого значення набувають цифрові ключі, які можуть зберігатися в мобільному застосунку, використовувати QR/NFC-ідентифікацію, підтримувати часові обмеження та бути пов’язаними з конкретними зонами доступу.

Система керування доступом до будівель з використанням цифрових ключів поєднує програмні модулі автентифікації, керування користувачами, видачі цифрових ключів, призначення прав доступу, взаємодії з терміналами доступу та журналювання подій. Таке рішення дає змогу адміністратору централізовано створювати, блокувати й відкликати ключі, охоронцю контролювати події входу та відмови, а користувачу отримувати зручний спосіб доступу до дозволених зон без використання фізичного носія.

Мета роботи полягає у розробленні програмного забезпечення системи керування доступом до будівель з використанням цифрових ключів, що забезпечує автентифікацію користувачів, видачу та перевірку цифрових ключів, керування правами доступу до зон, фіксацію подій і формування інформації для контролю безпеки об’єкта.

Для досягнення поставленої мети необхідно виконати такі завдання: проаналізувати предметну область систем контролю доступу; розглянути існуючі програмно-апаратні рішення для цифрового доступу; визначити функціональні та нефункціональні вимоги; розробити логічну і фізичну модель даних; побудувати UML-діаграми та алгоритми основних процесів; реалізувати програмні модулі системи; провести тестування та оцінити ефективність розробленого рішення.

Об’єктом дослідження є процес керування доступом користувачів до будівель, приміщень і зон безпеки з використанням цифрових ідентифікаторів.

Предметом дослідження є методи, моделі та програмні засоби реалізації системи цифрових ключів, прав доступу, автентифікації, журналювання подій і контролю доступу до приміщень.

Методи дослідження включають системний аналіз предметної області, UML-моделювання, ER-моделювання бази даних, структурне проєктування програмних модулів, алгоритмізацію основних сценаріїв роботи, методи тестування програмного забезпечення та оцінювання ефективності інформаційної системи.

Практичне значення роботи полягає у створенні програмного рішення, яке може бути використане як основа для інформаційної системи керування доступом у навчальному корпусі, офісній будівлі, гуртожитку або іншому об’єкті з поділом на зони доступу. Розроблена система забезпечує централізоване адміністрування цифрових ключів, контроль спроб входу, підвищення зручності роботи користувачів і посилення інформаційної підтримки служби безпеки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СИСТЕМИ КЕРУВАННЯ ДОСТУПОМ ДО БУДІВЕЛЬ З ВИКОРИСТАННЯМ ЦИФРОВИХ КЛЮЧІВ

1.1 Опис предметної області систем керування доступом до будівель

Предметна область систем керування доступом до будівель охоплює сукупність організаційних, апаратних і програмних засобів, призначених для ідентифікації користувачів, перевірки їхніх прав, надання або заборони входу до приміщень, а також фіксації подій доступу в журналі безпеки. У сучасних умовах традиційні механічні ключі та пластикові картки поступово доповнюються цифровими ключами, які можуть зберігатися у мобільному застосунку, електронному гаманці, QR-коді або іншому захищеному цифровому носії.

Цифровий ключ у такій системі розглядається як електронна облікова сутність, що містить інформацію про користувача, перелік дозволених точок доступу, строк дії, статус ключа та правила використання. На відміну від механічного ключа, цифровий ключ може бути швидко виданий, відкликаний, тимчасово заблокований або обмежений за часом, зоною будівлі чи роллю користувача. Це особливо важливо для офісних центрів, навчальних корпусів, житлових комплексів, готелів, складів, лабораторій і режимних приміщень.

Основними учасниками предметної області є користувач, адміністратор і охоронець. Користувач проходить автентифікацію, отримує цифровий ключ, подає запит на доступ або використовує ключ для входу до будівлі. Адміністратор створює облікові записи, керує правами доступу, блокує або відкликає цифрові ключі, переглядає журнал подій і налаштовує правила безпеки. Охоронець контролює події доступу, погоджує запити, реагує на підозрілі спроби входу та фіксує інциденти.

Система керування доступом функціонує як посередник між користувачем і фізичною інфраструктурою будівлі. До такої інфраструктури

можуть належати електронні замки, турнікети, зчитувачі NFC або BLE, контролери дверей, мобільні пристрої, сервер автентифікації та база даних. У процесі входу користувач пред'являє цифровий ключ, система перевіряє його чинність, права доступу та умови використання, після чого надсилає команду на відкриття або відмовляє у вході.

Для систематизації об'єктів предметної області у таблиці 1.1 наведено основні категорії сутностей, що використовуються під час розроблення інформаційної системи керування доступом до будівель з використанням цифрових ключів.

Таблиця 1.1

Основні категорії об'єктів предметної області системи керування доступом

Категорія об'єкта	Опис	Джерело даних
Користувач	Працівник, мешканець, гість або інша особа, яка отримує цифровий ключ і використовує його для входу	Обліковий запис користувача
Адміністратор	Відповідальна особа, що керує користувачами, правами доступу та цифровими ключами	Панель адміністрування
Охоронець	Працівник служби безпеки, який контролює події доступу та погоджує окремі запити	Журнал доступу, модуль спостереження
Цифровий ключ	Електронний ідентифікатор, який підтверджує право користувача на доступ до певної зони	Мобільний застосунок або сервер ключів
Точка доступу	Двері, турнікет, шлюз або інший об'єкт, через який здійснюється вхід	Довідник точок доступу
Подія доступу	Запис про успішну або відхилену спробу входу користувача	Журнал подій
Правило доступу	Умова, яка визначає час, зону, роль і допустимість входу	Модуль політик доступу

Сукупність наведених об'єктів формує інформаційну основу системи. Особливістю цифрового доступу є те, що право входу може змінюватися динамічно: користувачу можна надати тимчасовий ключ, обмежити доступ лише робочими годинами, відкрити одну зону будівлі або негайно відкликати ключ у разі втрати смартфона. Це підвищує гнучкість адміністрування та зменшує ризики, характерні для механічних ключів і незахищених карток.

1.2 Теоретико-методологічні засади використання цифрових ключів у системах доступу

Цифровий ключ у системі керування доступом є програмно сформованим ідентифікатором, який використовується для підтвердження права користувача на вхід до будівлі або окремого приміщення. На практиці цифровий ключ може реалізовуватися у вигляді мобільної облікової інформації, NFC- або BLE-ідентифікатора, QR-коду, тимчасового веб-посилання, PIN-коду або запису в електронному гаманці смартфона. Незалежно від конкретної технології, ключ повинен містити достатньо даних для перевірки автентичності, строку дії та прав доступу.

Технічно процес доступу складається з кількох етапів. Спочатку користувач проходить автентифікацію у мобільному застосунку або веб-інтерфейсі. Після цього сервер або локальний контролер перевіряє його роль, статус облікового запису та належність до групи доступу. Якщо користувач має дозвіл, йому видається цифровий ключ або активується вже виданий ключ. Під час входу ключ передається до зчитувача або контролера, який виконує перевірку та дозволяє відкриття дверей.

У системах цифрового доступу важливе значення має модель керування правами. Найчастіше використовується рольова модель, у якій права користувача визначаються його посадою або групою. Наприклад, працівник може мати доступ до основного входу та офісної зони, адміністратор - до серверної та службових приміщень, а гість - лише до переговорної кімнати в межах обмеженого часу. Для підвищення точності контролю також можуть застосовуватися атрибутивні правила: час доби, день тижня, статус ключа, геолокаційні умови або підтвердження охоронцем.

Критичною вимогою до цифрових ключів є захист від копіювання, несанкціонованої передачі та повторного використання. Для цього

застосовуються криптографічні механізми, прив’язка ключа до пристрою, обмеження строку дії, журналювання подій, двофакторна автентифікація та можливість негайного відкликання ключа. У разі втрати смартфона адміністратор може заблокувати цифровий ключ без фізичної заміни замків або перепрограмування всіх носіїв.

Основні способи реалізації цифрового ключа наведено у таблиці 1.2.

Таблиця 1.2

Способи реалізації цифрових ключів у системах доступу

Спосіб реалізації	Сутність технології	Переваги	Обмеження
NFC	Передавання ідентифікатора на малу відстань під час піднесення смартфона до зчитувача	Зручність і швидкість входу	Потреба у сумісному пристрої та зчитувачі
Bluetooth Low Energy	Передавання цифрового ключа між смартфоном і замком або контролером через BLE	Можливість безконтактного входу	Залежність від налаштувань смартфона та якості зв’язку
QR-код	Одноразовий або тимчасовий код для гостьового доступу	Просте надання доступу відвідувачам	Ризик пересилання коду стороннім особам
Мобільний застосунок	Зберігання ключа в обліковому записі користувача та керування входом через застосунок	Гнучке адміністрування і журналювання	Потреба у встановленні застосунку

З теоретичної точки зору система цифрового доступу повинна поєднувати ідентифікацію, автентифікацію, авторизацію та аудит. Ідентифікація встановлює, хто саме намагається отримати доступ. Автентифікація підтверджує достовірність користувача або пристрою. Авторизація визначає, чи має цей користувач право входу в конкретну зону. Аудит забезпечує збереження інформації про всі події доступу для подальшого аналізу та розслідування інцидентів.

1.3 Аналіз існуючих рішень й інформаційних систем

Ринок систем керування доступом активно розвивається у напрямі мобільних і цифрових ідентифікаторів. У сучасних рішеннях смартфон дедалі частіше використовується як заміна фізичної картки, брелока або механічного ключа. Для аналізу було обрано п’ять поширених рішень: HID Mobile Access, SALTO KS, Kisi Access Control, Brivo Mobile Pass та Avigilon Alta Access / Openpath.

HID Mobile Access є одним із відомих промислових рішень для мобільної ідентифікації. Воно використовує мобільний пристрій як обліковий носій для доступу до дверей, мереж і захищених сервісів. Перевагою рішення є орієнтація на корпоративний сегмент, підтримка захищених мобільних облікових даних і сумісність з інфраструктурою HID. Недоліком для невеликих об’єктів може бути складність впровадження та залежність від фірмового обладнання.

Рис. 1.1. Приклад мобільного доступу в HID Mobile Access

SALTO KS є хмарною системою контролю доступу, у якій цифровий ключ зберігається в мобільному застосунку та використовується для відкривання дверей через Bluetooth. Рішення підтримує керування замками, користувачами, правами доступу та гостьовими сценаріями. Сильними сторонами SALTO KS є хмарне адміністрування і підтримка різних типів електронних замків. Обмеженням є залежність від екосистеми SALTO та необхідність використання сумісного обладнання.

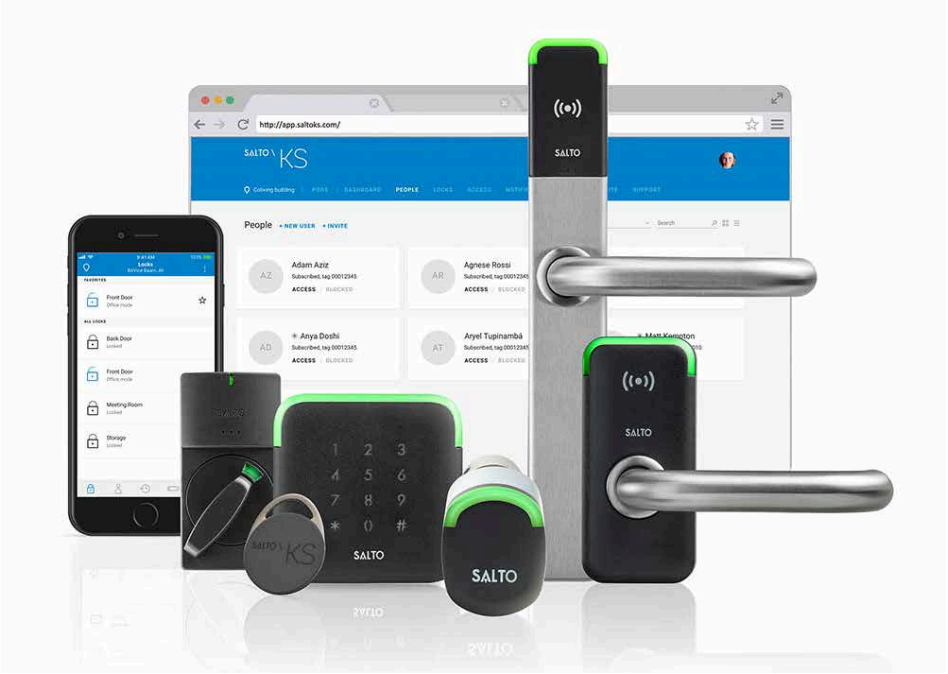

Рис. 1.2. Хмарна система керування доступом SALTO KS

Kisi Access Control орієнтована на хмарне керування доступом з використанням мобільних облікових даних, QR-кодів, веб-посилань і застосунку. Система підтримує тимчасовий доступ для відвідувачів і співробітників, а також дає змогу керувати дверима через веб-панель. Перевагою Kisi є зручність надання тимчасового доступу, однак використання посилань і QR-кодів потребує уважного контролю строку дії та ризиків пересилання доступу стороннім особам.

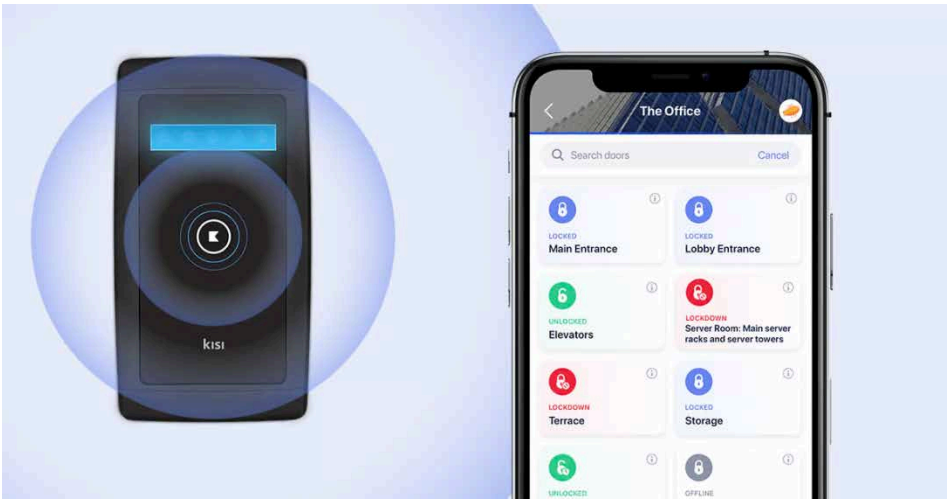

Рис. 1.3. Мобільний доступ і цифрові облікові дані у Kisi Access Control

Brivo Mobile Pass є рішенням для використання смартфона як електронної перепустки. Користувач може застосовувати мобільний пристрій для відкривання дверей, а адміністратор має змогу керувати доступом до різних об’єктів і зон. Перевагою Brivo є орієнтація на хмарне адміністрування будівельної безпеки та підтримка сценаріїв для офісів, житлових комплексів і комерційних об’єктів. Обмеженням є потреба у підключенні до платформи Brivo та сумісному апаратному забезпеченні.

Рис. 1.4. Використання смартфона як цифрового ключа у Brivo Mobile Pass

Avigilon Alta Access, що розвиває екосистему Openpath, є хмарною платформою контролю доступу з підтримкою мобільних облікових даних, централізованого адміністрування, інтеграцій із системами ідентифікації та

журналюванням подій. Рішення підтримує видачу мобільних облікових даних користувачам і може використовуватися для офісних будівель, навчальних закладів та корпоративних приміщень. Основною перевагою є масштабованість і розвинена панель керування, а обмеженням - залежність від хмарного сервісу та спеціалізованого обладнання.

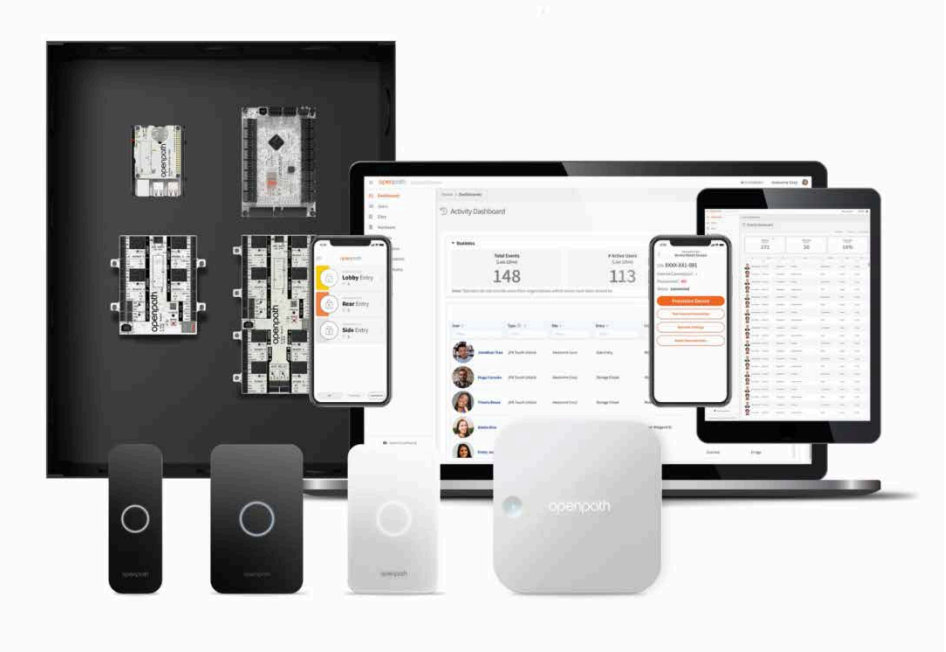

Рис. 1.5. Платформа Avigilon Alta Access / Openpath для керування доступом

Порівняльну характеристику розглянутих рішень наведено у таблиці 1.3.

Таблиця 1.3

Порівняння існуючих рішень із розроблюваною системою

Система	Цифровий ключ	Мобільний доступ	Тимчасовий доступ	Журнал подій	Обмеження
HID Mobile Access	Так	Так	Обмежено	Так	Залежність від фірмової інфраструктури HID
SALTO KS	Так	Так	Так	Так	Потреба у сумісних замках і хмарному сервісі
Kisi Access Control	Так	Так	Так	Так	Ризики пересилання цифрових посилань і QR-кодів

Продовження таблиці 1.3

Brivo Mobile Pass	Так	Так	Так	Так	Залежність від платформи Brivo
Avigilon Alta / Openpath	Так	Так	Так	Так	Орієнтація на комерційне апаратне середовище
Розроблювана система	Так	Так	Так	Так	Локальна навчальна реалізація з можливістю розширення

Загальний аналіз показує, що наявні промислові рішення мають розвинений функціонал, проте значна частина з них орієнтована на використання фірмового обладнання, хмарних сервісів і комерційних ліцензій. Для невеликих організацій, навчальних закладів або локальних будівель доцільним є створення програмного прототипу, який реалізує основну логіку цифрового доступу: автентифікацію, видачу ключів, погодження доступу, блокування ключів, перегляд журналу та контроль подій доступу.

1.4 Моделювання предметної області

Моделювання предметної області виконано з використанням UML-діаграм, які дозволяють формалізувати поведінку системи, ролі користувачів і послідовність виконання основних операцій. Для теми системи керування доступом до будівель з використанням цифрових ключів доцільно виділити трьох основних акторів: користувача, адміністратора та охоронця.

Користувач виконує автентифікацію, отримує цифровий ключ, використовує його для входу та за потреби подає запит на доступ. Адміністратор керує правами доступу, блокує або відкликає ключі, а також переглядає журнал доступу. Охоронець контролює події доступу, погоджує

окремі запити та фіксує факти входу до будівлі. Діаграму прецедентів системи подано на рисунку 1.6.

Рис. 1.6. Діаграма прецедентів системи керування доступом до будівель

Діаграма прецедентів демонструє, що центральними функціями системи є автентифікація, отримання цифрового ключа, використання ключа для входу, подання запиту на доступ, керування правами, блокування або відкликання ключа, перегляд журналу та контроль подій доступу. Така структура відповідає типовому сценарію експлуатації системи в офісній або адміністративній будівлі.

Для деталізації процесу видачі та використання цифрового ключа побудовано діаграму послідовності, наведену на рисунку 1.7. Вона відображає взаємодію між користувачем, охоронцем і адміністратором під час подання запиту на доступ, погодження прав і подальшого входу до будівлі.

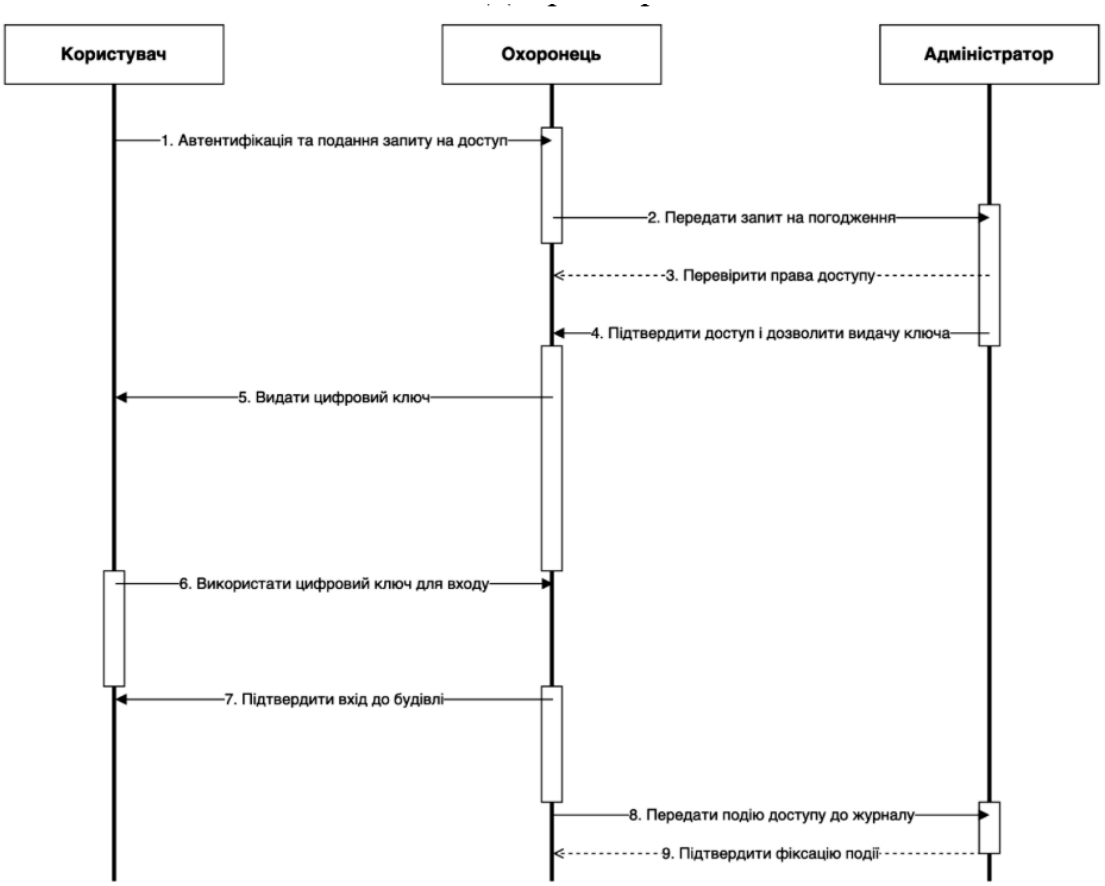

Рис. 1.7. Діаграма послідовності видачі та використання цифрового ключа

Сценарій починається з автентифікації користувача та подання запиту на доступ. Охоронець приймає запит і передає його адміністратору для перевірки прав. Адміністратор аналізує права доступу, після чого підтверджує або відхиляє запит. У разі підтвердження охоронець видає користувачу цифровий ключ. Користувач застосовує ключ для входу, а система фіксує подію в журналі доступу.

Логіка роботи процесу у розрізі ролей подана на діаграмі активностей, що наведена на рисунку 1.8. Діаграма містить доріжки користувача, охоронця й адміністратора, а також показує розгалуження сценарію залежно від результату перевірки прав доступу.

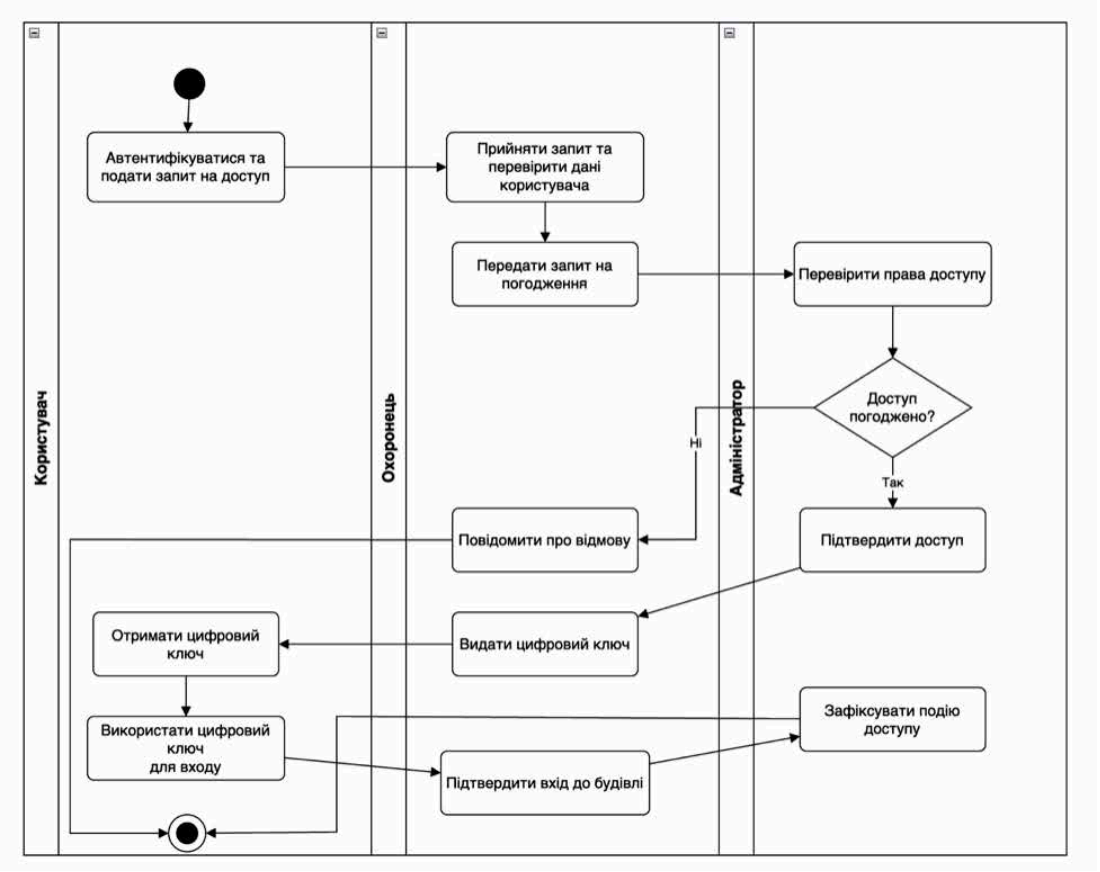

Рис. 1.8. Діаграма активностей процесу отримання та використання цифрового ключа

На діаграмі активностей видно, що система передбачає два основні варіанти розвитку подій. Якщо доступ погоджено, адміністратор підтверджує запит, охоронець видає цифровий ключ, користувач використовує його для входу, а подія доступу фіксується в журналі. Якщо доступ не погоджено, користувач отримує повідомлення про відмову. Такий підхід дозволяє формалізувати правила доступу й забезпечити контроль усіх дій у системі.

1.5 Аналіз вимог інформаційної системи

Аналіз вимог до системи керування доступом до будівель з використанням цифрових ключів передбачає визначення функціональних, нефункціональних і безпекових характеристик майбутнього програмного забезпечення. Вимоги сформовано з урахуванням потреб користувача,

адміністратора та охоронця, а також особливостей роботи з цифровими ключами.

Функціональні вимоги визначають основні можливості системи, які мають бути реалізовані для забезпечення повного циклу керування доступом. Перелік функціональних вимог наведено у таблиці 1.4.

Таблиця 1.4

Функціональні вимоги до системи керування доступом

№	Вимога	Опис
1	Автентифікація користувача	Перевірка логіна, пароля або іншого способу підтвердження особи
2	Видача цифрового ключа	Створення цифрового ключа для користувача з урахуванням строку дії та дозволених зон
3	Використання ключа для входу	Перевірка ключа під час спроби входу до будівлі або приміщення
4	Подання запиту на доступ	Формування користувачем запиту на отримання або розширення прав доступу
5	Погодження доступу	Перевірка прав користувача адміністратором або охоронцем
6	Керування правами доступу	Призначення ролей, зон, часових обмежень і статусів ключів
7	Блокування та відкликання ключа	Деактивація ключа у разі втрати пристрою, звільнення користувача або завершення строку дії
8	Журнал подій доступу	Фіксація успішних і відхилених спроб входу з датою, часом і користувачем
9	Контроль подій охоронцем	Перегляд поточних подій, реагування на підозрілі спроби доступу
10	Експорт звітів	Формування звітної інформації за користувачами, ключами та подіями доступу

Нефункціональні вимоги характеризують якість роботи системи, її зручність, швидкодію, надійність і можливість подальшого масштабування. Їх наведено у таблиці 1.5.

Таблиця 1.5

Нефункціональні вимоги до інформаційної системи

№	Вимога	Показник або характеристика
1	Швидкодія	Перевірка цифрового ключа та відкриття доступу мають виконуватися без помітної затримки для користувача
2	Надійність	Система має зберігати працездатність при типових помилках введення та повторних запитах

3	Зручність інтерфейсу	Основні дії мають виконуватися через зрозумілі форми та таблиці
4	Масштабованість	Структура системи має дозволяти додавання нових будівель, дверей і користувачів
5	Супровідність	Програмний код має бути модульним і придатним для подальшого розширення
6	Сумісність	Система має підтримувати інтеграцію з різними типами цифрових ключів і пристроїв доступу
7	Журналювання	Усі критичні дії мають фіксуватися для подальшого аналізу

Вимоги до безпеки є ключовими, оскільки система обробляє дані користувачів і керує фізичним доступом до будівель. Основні безпекові вимоги наведено у таблиці 1.6.

Таблиця 1.6

Вимоги до безпеки системи керування доступом

№	Вимога	Опис
1	Захист облікових даних	Паролі та ключові дані не зберігаються у відкритому вигляді
2	Розмежування ролей	Користувач, охоронець і адміністратор мають різні права
3	Строк дії ключів	Цифрові ключі можуть бути постійними, тимчасовими або одноразовими
4	Відкликання ключів	Адміністратор може негайно заблокувати ключ

Продовження таблиці 1.6

5	Контроль підозрілих подій	Повторні відмови або спроби входу в заборонену зону фіксуються в журналі
6	Аудит дій	Операції видачі, зміни, блокування і використання ключів зберігаються в журналі
7	Валідація вхідних даних	Система перевіряє коректність введених значень і не допускає збереження помилкових записів

Узагальнення вимог показує, що система має поєднувати функції автентифікації, керування цифровими ключами, адміністрування прав доступу, контролю подій та аудиту. Такий набір вимог є достатнім для подальшого проєктування структури бази даних, програмних модулів і користувацького інтерфейсу.

1.6 Постановка завдання

Постановка завдання для розроблення системи керування доступом до будівель з використанням цифрових ключів ґрунтується на результатах аналізу предметної області, існуючих рішень, UML-моделей і сформованих вимог. Основне завдання полягає у створенні програмного забезпечення, яке забезпечує контрольований доступ користувачів до будівлі або окремих приміщень за допомогою цифрових ключів.

Вхідними даними системи є облікові дані користувачів, ролі, перелік будівель і точок доступу, параметри цифрових ключів, правила доступу, запити користувачів, рішення адміністратора або охоронця, а також події використання ключів. Кожен цифровий ключ має містити інформацію про власника, дозволені зони, строк дії, статус і спосіб використання.

Вихідними даними системи є видані цифрові ключі, підтвердження або відмова у доступі, записи журналу подій, повідомлення для користувачів, звіти про використання ключів, інформація про заблоковані або відкликані ключі, а також аналітичні дані для адміністратора та охоронця.

У межах розроблення необхідно реалізувати модуль автентифікації користувачів, модуль керування цифровими ключами, модуль подання й погодження запитів на доступ, модуль адміністрування прав, модуль журналу подій і модуль формування звітів. Окремо потрібно передбачити оброблення ситуацій відмови у доступі, відкликання ключа та фіксації підозрілих дій.

Результатом виконання завдання має стати програмна система, яка моделює повний цикл керування цифровим доступом: від створення користувача й видачі цифрового ключа до фактичного використання ключа для входу та запису події в журналі. Система повинна бути придатною для демонстрації основних принципів цифрового доступу, подальшого розширення та використання як основи для кваліфікаційної роботи.

Таким чином, розроблюване програмне забезпечення має забезпечити підвищення контрольованості доступу до будівель, зменшення залежності від фізичних ключів, спрощення адміністрування прав користувачів і створення надійного журналу подій для аналізу безпеки.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель даних системи керування доступом до будівель з використанням цифрових ключів

Логічна модель даних визначає інформаційну основу системи керування доступом до будівель з використанням цифрових ключів. Вона описує основні сутності предметної області, їх атрибути, первинні та зовнішні ключі, а також зв’язки між користувачами, цифровими ключами, зонами доступу, пристроями контролю та подіями входу. Побудова такої моделі є необхідною передумовою для подальшого створення фізичної структури бази даних і програмних модулів системи.

У межах проєктованої системи дані мають зберігатися у структурованому вигляді, щоб забезпечити облік користувачів, керування їхніми ролями, видачу цифрових ключів, визначення дозволів на доступ до конкретних зон будівлі, реєстрацію спроб входу та контроль подій безпеки. Логічна модель враховує те, що один користувач може мати одну роль, але в межах цієї ролі отримувати декілька цифрових ключів або дозволів доступу для різних зон будівлі.

На рис. 2.1 наведено логічну модель даних системи. Вона включає сутності ROLE, USER, DIGITAL_KEY, ACCESS_PERMISSION, BUILDING, ACCESS_ZONE, ACCESS_DEVICE та ACCESS_EVENT. Такий набір сутностей дозволяє описати повний цикл роботи системи: від створення користувача і видачі цифрового ключа до перевірки дозволу та фіксації події доступу.

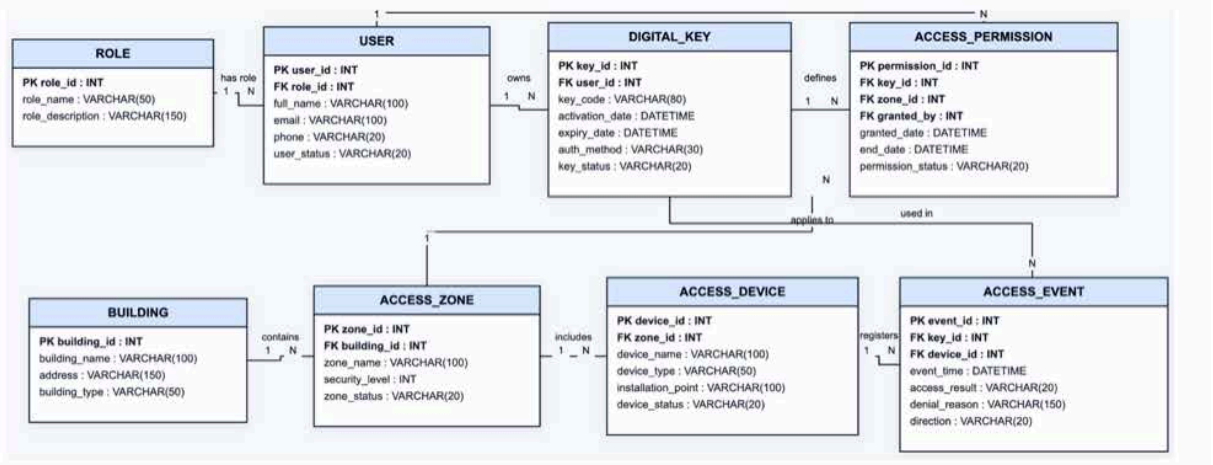

Рис. 2.1. Логічна модель даних системи керування доступом до будівель

Сутність ROLE зберігає типи користувачів системи. До таких ролей можуть належати адміністратор, охоронець, співробітник, тимчасовий відвідувач або службовий персонал. Зв’язок між ROLE та USER має тип один-до-багатьох, оскільки одна роль може бути призначена багатьом користувачам, але кожен користувач у базовій моделі має одну основну роль.

Сутність USER містить персональні та службові дані користувача: повне ім’я, електронну пошту, телефон, статус облікового запису та посилання на роль. USER пов’язується із DIGITAL_KEY, оскільки один користувач може мати один або декілька цифрових ключів. Це важливо для підтримки різних сценаріїв доступу, наприклад постійного ключа для співробітника та тимчасового ключа для відвідувача.

Сутність DIGITAL_KEY описує цифровий ключ, який використовується для входу до будівлі або окремої зони. У ній зберігається код ключа, дата активації, дата завершення дії, метод автентифікації та статус ключа. Статус може набувати значень активний, заблокований, відкликаний або прострочений. Ці дані дозволяють обмежувати доступ без фізичної заміни ключа або картки.

Сутність ACCESS_PERMISSION визначає, до яких зон застосовується конкретний цифровий ключ. Вона пов’язує ключ із зоною доступу, містить дату надання дозволу, дату завершення дії дозволу та його статус. Це дає змогу

гнучко керувати правами доступу і надавати користувачу вхід лише до тих приміщень, які відповідають його ролі або службовому завданню.

Сутності BUILDING та ACCESS_ZONE використовуються для опису просторової структури об’єкта. Одна будівля може містити декілька зон доступу: головний вхід, офісну частину, серверну кімнату, складське приміщення або технічну зону. ACCESS_DEVICE пов’язується із зоною доступу і описує конкретний пристрій зчитування, наприклад NFC-термінал, BLE-зчитувач або QR-сканер.

Сутність ACCESS_EVENT зберігає фактичні події доступу. Кожна подія пов’язується з цифровим ключем і пристроєм доступу. У записі зберігається час події, результат перевірки, причина відмови та напрям руху. Це забезпечує аудит входів і виходів, виявлення несанкціонованих спроб доступу та формування журналу подій для адміністратора або охоронця.

Таблиця 2.1

Основні сутності логічної моделі даних

Сутність	Призначення	Основні атрибути
ROLE	Зберігання ролей користувачів системи	role_id, role_name, role_description
USER	Облік користувачів, які отримують цифрові ключі	user_id, role_id, full_name, email, phone, user_status
DIGITAL_KEY	Зберігання цифрових ключів і параметрів їх дії	key_id, user_id, key_code, activation_date, expiry_date, auth_method, key_status
ACCESS_PERMISSION	Визначення прав доступу цифрового ключа до зон будівлі	permission_id, key_id, zone_id, granted_by, granted_date, end_date, permission_status
BUILDING	Опис будівель, у яких діє система доступу	building_id, building_name, address, building_type
ACCESS_ZONE	Опис зон доступу всередині будівлі	zone_id, building_id, zone_name, security_level, zone_status
ACCESS_DEVICE	Облік пристроїв зчитування цифрових ключів	device_id, zone_id, device_name, device_type, installation_point, device_status
ACCESS_EVENT	Фіксація подій входу, відмови або виходу	event_id, key_id, device_id, event_time, access_result, denial_reason, direction

Логічна модель побудована з урахуванням принципів нормалізації. Дані про користувача не дублюються в таблиці цифрових ключів, а зберігаються

окремо і пов’язуються через зовнішній ключ. Аналогічно, події доступу не містять повного опису зони чи користувача, а посилаються на відповідний ключ і пристрій. Це зменшує надлишковість інформації та забезпечує цілісність даних.

Запропонована модель є достатньою для подальшої реалізації бази даних системи. Вона підтримує основні функції: створення користувачів, видачу ключів, блокування або відкликання ключів, призначення прав доступу, ведення журналу входів і контроль пристроїв доступу в різних зонах будівлі.

2.2 Діаграма класів і кооперації системи керування доступом

Діаграма класів відображає об’єктно-орієнтовану структуру програмного забезпечення системи керування доступом. На відміну від логічної моделі даних, яка описує збереження інформації у базі, діаграма класів показує програмні об’єкти, їх властивості, методи та зв’язки, що використовуються під час реалізації функціоналу цифрових ключів.

Основними класами системи є Користувач, ЦифровийКлюч, ПравоДоступу, Будівля, ЗонаДоступу, ПристрійДоступу та ПодіяДоступу. Вони охоплюють ключові частини предметної області: суб’єкт доступу, цифровий ідентифікатор, набір дозволів, об’єкт доступу, технічний пристрій і результат перевірки входу. Діаграму класів наведено на рис. 2.2.

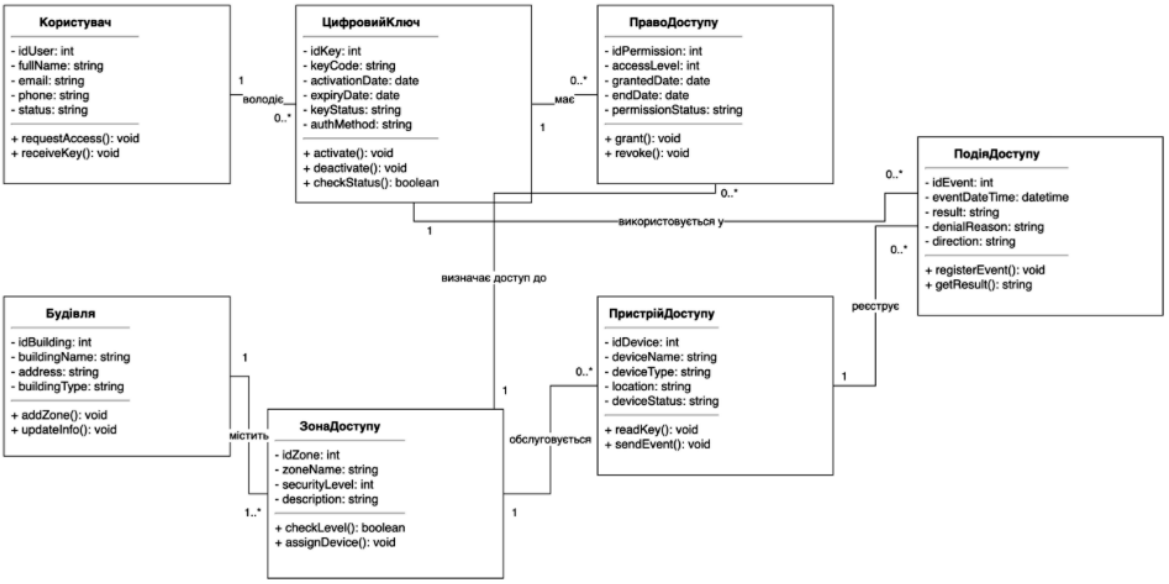

Рис. 2.2. Діаграма класів системи керування доступом до будівель

Клас Користувач містить ідентифікатор, повне ім’я, електронну пошту, номер телефону та статус. Його основні методи пов’язані з поданням запиту на доступ і отриманням цифрового ключа. Клас ЦифровийКлюч містить код ключа, дату активації, дату завершення дії, статус і метод автентифікації. Методи цього класу забезпечують активацію, деактивацію та перевірку статусу ключа.

Клас ПравоДоступу визначає межі використання цифрового ключа. Він містить рівень доступу, дату надання дозволу, дату завершення дії дозволу та статус. Методи grant() і revoke() використовуються для надання або відкликання права доступу. Клас ЗонаДоступу описує зону будівлі, до якої може бути надано доступ, а клас ПристрійДоступу відповідає за зчитування ключа та передавання події до системи.

Клас ПодіяДоступу використовується для збереження результату фактичної спроби входу. Він містить час події, результат перевірки, причину відмови та напрям руху. Цей клас важливий для аудиту безпеки, оскільки саме на основі подій формується журнал доступу.

Таблиця 2.2

Характеристика основних класів програмної системи

Клас	Основне призначення	Ключові методи
Користувач	Подання запиту на доступ і отримання цифрового ключа	requestAccess(), receiveKey()
ЦифровийКлюч	Зберігання та перевірка цифрового ідентифікатора доступу	activate(), deactivate(), checkStatus()
ПравоДоступу	Визначення дозволів для конкретного ключа	grant(), revoke()
Будівля	Опис об’єкта, у якому використовуються зони доступу	addZone(), updateInfo()
ЗонаДоступу	Керування рівнем безпеки та пристроями зони	checkLevel(), assignDevice()
ПристрійДоступу	Зчитування ключа та передавання події доступу	readKey(), sendEvent()
ПодіяДоступу	Реєстрація результату спроби входу	registerEvent(), getResult()

Для деталізації взаємодії класів у системі побудовано кооперації. Вони показують, як програмні об’єкти взаємодіють між собою під час виконання окремих сценаріїв. У роботі розглянуто три основні кооперації: видачу цифрового ключа, використання цифрового ключа для входу та реєстрацію події доступу.

Перша кооперація описує сценарій видачі цифрового ключа. Користувач подає запит на доступ, охоронець перевіряє запит і передає його адміністратору на погодження. Адміністратор підтверджує доступ, після чого формується цифровий ключ і передається користувачу. Цей сценарій наведено на рис. 2.3.

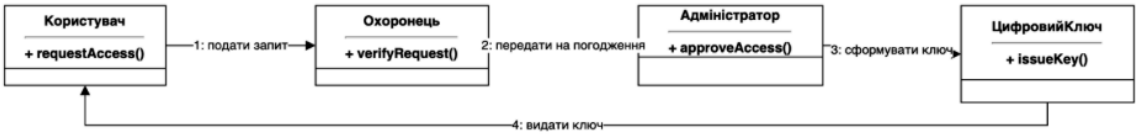

Рис. 2.3. Кооперація процесу видачі цифрового ключа

Друга кооперація відображає процес використання цифрового ключа для входу. Користувач застосовує цифровий ключ, ключ передає код пристрою доступу, пристрій надсилає запит на перевірку права доступу, після чого система дозволяє або забороняє вхід. Така взаємодія є основною для щоденної експлуатації системи та наведена на рис. 2.4.

Рис. 2.4. Кооперація процесу використання цифрового ключа

Третя кооперація описує процес реєстрації події доступу. Пристрій доступу створює подію, об’єкт ПодіяДоступу реєструє результат, після чого інформація пов’язується із зоною доступу і стає доступною адміністратору через журнал. Такий механізм дозволяє відстежувати всі спроби входу, включно з відмовами, та оперативно аналізувати інциденти. Кооперацію наведено на рис. 2.5.

Рис. 2.5. Кооперація процесу реєстрації події доступу

Кооперації підтверджують, що основні сценарії системи реалізуються через узгоджену взаємодію об’єктів. Видача ключа передбачає участь користувача, охоронця, адміністратора та сервісу формування ключів. Використання ключа базується на взаємодії користувача, цифрового ключа, пристрою доступу і механізму перевірки дозволів. Реєстрація події забезпечує аудит дій і формує інформаційну основу для контролю безпеки.

2.3 Діаграма компонентів програмного забезпечення

Діаграма компонентів відображає архітектурну структуру програмного забезпечення системи керування доступом. Вона показує основні програмні модулі, канали взаємодії між ними та зв’язок із базою даних. Для проєктованої системи доцільною є модульна архітектура, у якій клієнтські інтерфейси взаємодіють із серверними сервісами через API-шлюз.

На рис. 2.6 наведено компонентну модель системи. Вона включає веб-інтерфейс адміністратора, мобільний клієнт користувача, API-шлюз, сервіс автентифікації, сервіс керування цифровими ключами, сервіс керування доступом, сервіс журналювання подій, модуль доступу до даних, сервіс сповіщень та базу даних AccessControlDB.

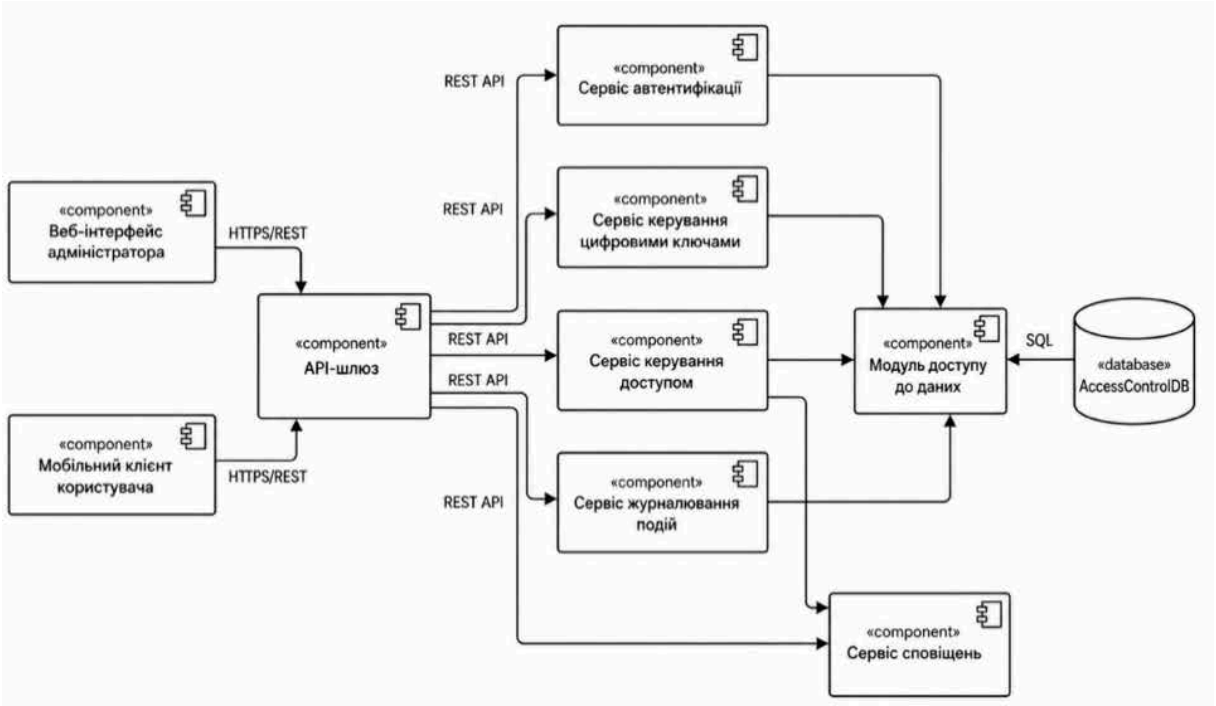

Рис. 2.6. Діаграма компонентів системи керування доступом

Веб-інтерфейс адміністратора призначений для створення користувачів, керування ролями, видачі ключів, призначення дозволів і перегляду журналу доступу. Мобільний клієнт користувача використовується для отримання цифрового ключа, подання запиту на доступ і застосування ключа під час входу до будівлі або зони доступу.

API-шлюз є центральною точкою взаємодії клієнтських застосунків із серверною частиною. Він приймає HTTPS/REST-запити, виконує маршрутизацію до відповідних сервісів і забезпечує відокремлення клієнтського рівня від внутрішньої логіки системи. Через API-шлюз проходять запити на автентифікацію, керування ключами, перевірку дозволів і отримання журналу подій.

Сервіс автентифікації відповідає за перевірку облікових даних користувачів і підтвердження їхньої ролі. Сервіс керування цифровими ключами забезпечує створення, активацію, блокування та відкликання ключів. Сервіс керування доступом перевіряє, чи має конкретний ключ право входу до вибраної зони. Сервіс журналювання подій фіксує результати спроб доступу та передає їх до бази даних.

Модуль доступу до даних ізолює роботу бізнес-логіки від конкретної реалізації сховища. Усі основні сервіси звертаються до бази AccessControlDB через цей модуль. Такий підхід спрощує супровід системи, оскільки в разі зміни СУБД або структури сховища не потрібно переписувати всі програмні компоненти.

Таблиця 2.3

Призначення компонентів програмного забезпечення

Компонент	Призначення
Веб-інтерфейс адміністратора	Керування користувачами, цифровими ключами, зонами доступу та журналом подій
Мобільний клієнт користувача	Отримання та використання цифрового ключа для входу до будівлі
API-шлюз	Маршрутизація запитів між клієнтськими застосунками та серверними сервісами
Сервіс автентифікації	Перевірка користувача, ролі та права входу до системи
Сервіс керування цифровими ключами	Видача, активація, блокування і відкликання цифрових ключів
Сервіс керування доступом	Перевірка дозволів на вхід до конкретної зони будівлі
Сервіс журналювання подій	Фіксація спроб входу, відмов, часу події та результату перевірки
Сервіс сповіщень	Надсилання повідомлень користувачам, охоронцю або адміністратору
Модуль доступу до даних	Виконання запитів до бази даних та повернення результатів сервісам
AccessControlDB	Зберігання користувачів, ключів, дозволів, зон, пристроїв та подій доступу

Запропонована компонентна структура забезпечує розділення відповідальності між окремими частинами системи. Це підвищує надійність програмного забезпечення, полегшує тестування модулів і створює основу для подальшого розширення системи, наприклад додавання біометричної автентифікації, інтеграції з відеоспостереженням або підтримки гостьових цифрових ключів.

2.4 Діаграма пакетів системи керування доступом

Діаграма пакетів використовується для подання логічної організації програмного коду. Вона групує споріднені класи, сервіси та модулі у пакети, які відповідають окремим функціональним частинам системи. Такий підхід спрощує розробку, супровід і розширення програмного забезпечення.

У проєктованій системі пакети поділено на клієнтський і серверний рівні. Клієнтський рівень містить пакети UserInterface, AccessRequest та Notification. Серверний рівень містить пакети AuthAccess, KeyManagement, EventJournal та DataAccess. Діаграму пакетів наведено на рис. 2.7.

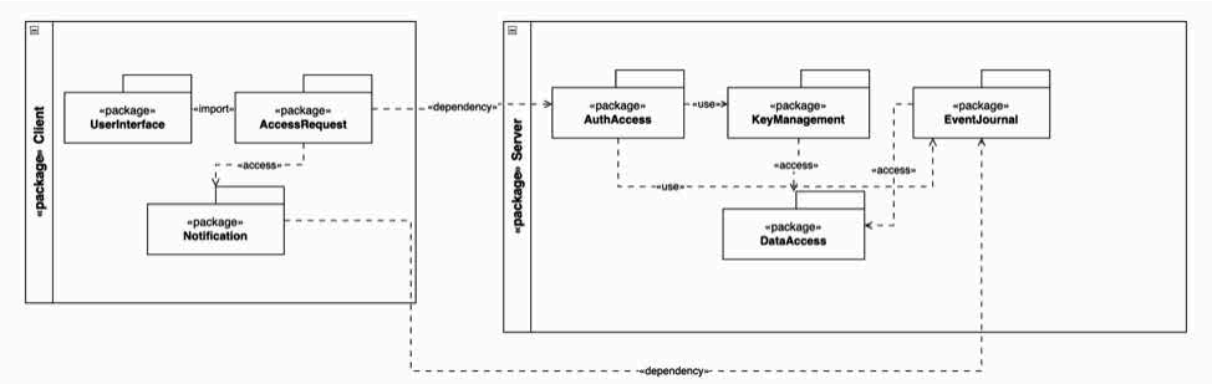

Рис. 2.7. Діаграма пакетів системи керування доступом

Пакет UserInterface містить елементи інтерфейсу користувача та адміністратора. Через нього користувач подає запит на доступ, переглядає доступні цифрові ключі та отримує повідомлення. Пакет AccessRequest відповідає за створення й передавання запитів на доступ. Пакет Notification

використовується для отримання повідомлень про результат погодження, блокування ключа або подію доступу.

На серверному рівні пакет AuthAccess відповідає за автентифікацію і перевірку прав користувача. Пакет KeyManagement містить логіку створення, оновлення, блокування та відкликання цифрових ключів. Пакет EventJournal забезпечує запис подій доступу та формування журналу. Пакет DataAccess реалізує доступ до бази даних і використовується іншими серверними пакетами для зчитування або збереження інформації.

Залежності між пакетами показують напрям використання функцій. Клієнтські пакети звертаються до серверних через прикладний інтерфейс, а серверні пакети використовують DataAccess для роботи з базою даних. Пакет EventJournal пов’язаний із процесами доступу, оскільки кожна перевірка ключа або спроба входу має фіксуватися в журналі.

Таблиця 2.4

Характеристика пакетів програмного забезпечення

Пакет	**Рівень**	**Призначення**
UserInterface	Клієнтський	Форми й екрани взаємодії користувача та адміністратора із системою
AccessRequest	Клієнтський	Створення, передавання та відстеження запитів на доступ
Notification	Клієнтський	Отримання повідомлень про видані ключі, відмови та події безпеки
AuthAccess	Серверний	Автентифікація користувача та перевірка ролі
KeyManagement	Серверний	Керування цифровими ключами протягом їх життєвого циклу
EventJournal	Серверний	Запис і перегляд журналу подій доступу
DataAccess	Серверний	Уніфікований доступ до даних системи

Пакетна структура є достатньо компактною і водночас відображає основні межі відповідальності програмних модулів. Вона дозволяє розділити клієнтський інтерфейс, оброблення запитів, логіку автентифікації, керування

ключами, журналювання подій і доступ до даних. Надалі така структура може бути розширена пакетами для інтеграції з апаратними контролерами, біометричними модулями або зовнішніми сервісами безпеки.

2.5 Висновки до другого розділу

У другому розділі виконано проєктування інформаційного та програмного забезпечення системи керування доступом до будівель з використанням цифрових ключів. Розроблено логічну модель даних, яка включає користувачів, ролі, цифрові ключі, дозволи доступу, будівлі, зони доступу, пристрої та події доступу. Визначені зв’язки між сутностями забезпечують цілісне зберігання даних і підтримують основні сценарії функціонування системи.

Побудовано діаграму класів, яка відображає об’єктно-орієнтовану структуру програмного забезпечення. Визначено основні класи, їх атрибути, методи та взаємозв’язки. Додатково розроблено кооперації для процесів видачі цифрового ключа, використання ключа для входу та реєстрації події доступу. Ці кооперації деталізують послідовність взаємодії програмних об’єктів у ключових сценаріях роботи системи.

Розроблено компонентну модель програмного забезпечення, яка передбачає взаємодію веб-інтерфейсу адміністратора, мобільного клієнта користувача, API-шлюзу, сервісів автентифікації, керування ключами, керування доступом, журналювання подій, сповіщень і бази даних AccessControlDB. Така структура забезпечує модульність, масштабованість і можливість подальшого розширення системи.

Також побудовано діаграму пакетів, яка відображає логічну організацію програмного коду на клієнтському та серверному рівнях. Запропонована структура дозволяє чітко розділити інтерфейс, оброблення запитів, автентифікацію, керування цифровими ключами, журналювання подій і доступ

до даних. Отримані результати проєктування створюють основу для подальшої реалізації бази даних, програмних модулів та інтерфейсу системи.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Обґрунтування вибору технологічних засобів реалізації

Розроблення програмного забезпечення системи керування доступом до будівель з використанням цифрових ключів передбачає створення прикладного рішення, яке забезпечує автентифікацію користувачів, видачу цифрових ключів, перевірку прав доступу, керування зонами будівлі, облік пристроїв доступу та журналювання подій входу. З огляду на це технологічні засоби мають забезпечувати надійне збереження даних, захищену взаємодію між клієнтською і серверною частинами, можливість масштабування та зручність подальшого супроводу.

Архітектура програмного забезпечення орієнтована на клієнт-серверну модель. На клієнтському рівні передбачено веб-інтерфейс адміністратора та мобільний інтерфейс користувача. Веб-інтерфейс використовується для керування користувачами, цифровими ключами, правами доступу, зонами й журналом подій. Мобільний клієнт забезпечує отримання цифрового ключа, його зберігання та використання під час входу до будівлі або окремої зони доступу.

Серверна частина відповідає за бізнес-логіку системи: перевірку облікових даних, формування цифрових ключів, призначення дозволів, оброблення подій від пристроїв доступу та збереження інформації в базі даних. Для реалізації серверної логіки доцільно застосувати мову Python і фреймворк FastAPI, оскільки вони дозволяють створити REST API, мають підтримку валідації даних і добре підходять для побудови модульних інформаційних систем.

Для збереження даних про користувачів, ключі, зони, пристрої та події доступу використовується реляційна база даних AccessControlDB. У межах

прототипу структура бази даних може бути реалізована у SQLite або PostgreSQL. SQLite є зручним для локального тестування й демонстрації, а PostgreSQL доцільний для промислового розгортання, оскільки підтримує цілісність даних, транзакції, обмеження та роботу з великим обсягом журналів доступу.

Передача даних між клієнтськими застосунками та серверною частиною здійснюється через HTTPS/REST. Для захисту сеансів користувачів передбачено токенну автентифікацію. Цифровий ключ у системі розглядається як програмний ідентифікатор доступу, який має власний код, строк дії, метод автентифікації та статус. Такий підхід дає змогу швидко видавати, блокувати або відкликати ключ без фізичної заміни картки чи брелока.

Таблиця 3.1

Технологічні засоби реалізації системи

Засіб	Призначення у системі
Python	Реалізація серверної логіки, алгоритмів перевірки доступу та оброблення подій
FastAPI / REST API	Створення прикладного інтерфейсу для взаємодії веб-інтерфейсу, мобільного клієнта та пристроїв доступу
SQLite / PostgreSQL	Збереження користувачів, ролей, цифрових ключів, дозволів, зон, пристроїв і подій доступу
HTML, CSS, JavaScript	Побудова адміністративного інтерфейсу для керування доступом
JWT-токени	Підтвердження автентичності користувача під час роботи з API
HTTPS	Захищена передача даних між клієнтами та сервером
JSON	Формат обміну даними між клієнтськими застосунками та серверною частиною

Використання зазначених технологій дозволяє реалізувати систему як сукупність пов’язаних програмних модулів. Такий підхід забезпечує поділ відповідальності між клієнтським інтерфейсом, сервісом автентифікації, сервісом керування цифровими ключами, сервісом перевірки прав доступу, модулем журналювання та базою даних. У результаті система може бути розширена без повного переписування програмного коду.

Важливою вимогою до системи є контроль цілісності даних. Тому структура бази даних має містити первинні та зовнішні ключі, унікальні обмеження, статуси записів і часові мітки. Це дозволяє відстежувати життєвий цикл цифрового ключа від моменту створення до блокування або завершення строку дії.

3.2 Розробка фізичної моделі бази даних

Фізична модель бази даних деталізує структуру збереження даних системи керування доступом до будівель з використанням цифрових ключів. На відміну від логічної моделі, фізична модель визначає назви таблиць, типи полів, первинні ключі, зовнішні ключі, обмеження цілісності та можливі значення статусів. Така модель є основою для створення бази даних AccessControlDB.

До складу фізичної моделі входять таблиці ROLES, USERS, DIGITAL_KEYS, ACCESS_PERMISSIONS, BUILDINGS, ACCESS_ZONES, ACCESS_DEVICES та ACCESS_EVENTS. Вони забезпечують збереження ролей користувачів, персональних даних користувачів, цифрових ключів, прав доступу, будівель, зон доступу, пристроїв зчитування та фактичних подій входу. Фізичну модель бази даних наведено на рис. 3.1.

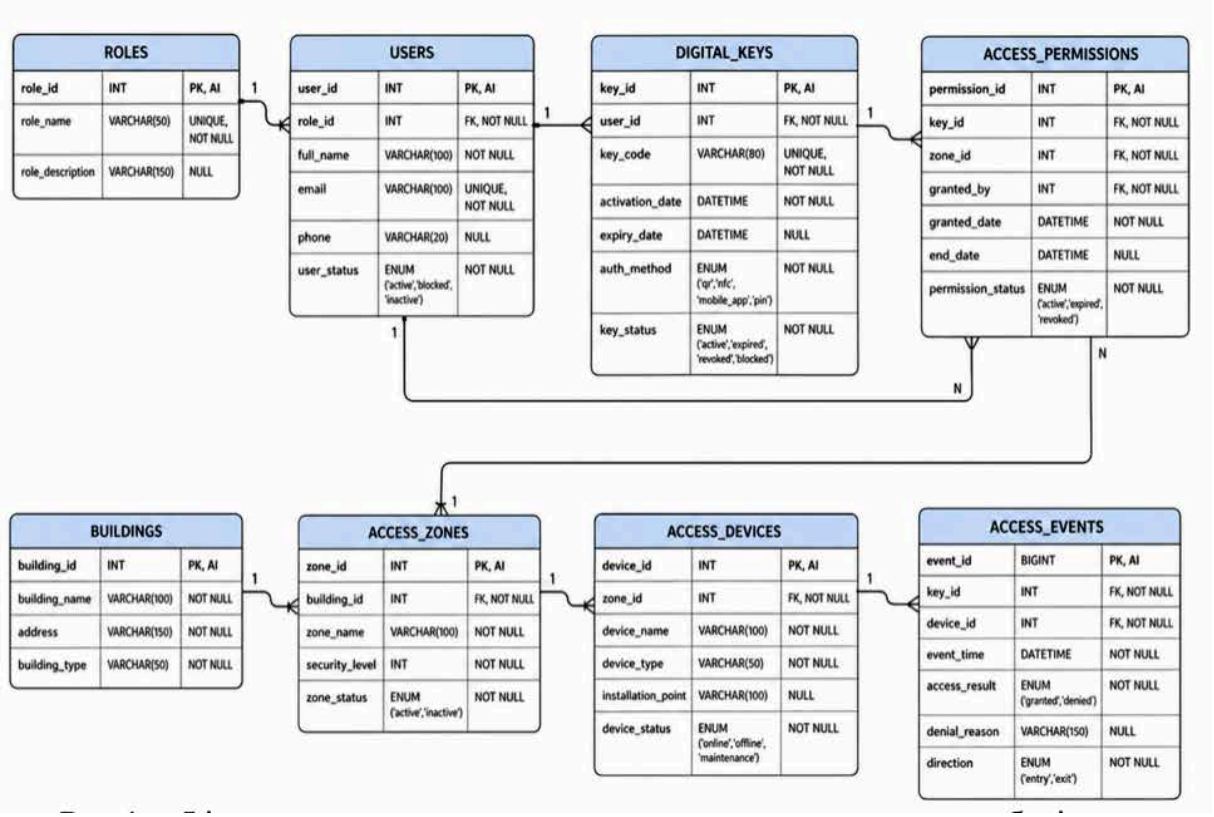

Рис. 3.1. Фізична модель бази даних системи керування доступом

Таблиця ROLES використовується для збереження ролей користувачів. Вона містить ідентифікатор ролі, назву ролі та опис. Через таблицю USERS роль призначається конкретному користувачу. Така структура дозволяє відокремити службову роль користувача від його персональних даних і використовувати одну роль для багатьох користувачів.

Таблиця USERS зберігає облікові дані користувачів системи. До неї входять ідентифікатор користувача, посилання на роль, повне ім’я, електронна пошта, телефон і статус користувача. Статус дає змогу заблокувати користувача без видалення його історії доступу. Це важливо для збереження журналу дій та подій входу.

Таблиця DIGITAL_KEYS описує цифрові ключі. Для кожного ключа зберігається код, дата активації, дата завершення дії, метод автентифікації та статус. Зв’язок із USERS показує, кому належить конкретний ключ. Унікальність key_code запобігає створенню двох однакових ключів у системі.

Таблиця ACCESS_PERMISSIONS визначає права доступу цифрового ключа до конкретних зон. Вона пов’язує ключ із зоною доступу, зберігає адміністратора, який надав дозвіл, дату надання, дату завершення дії дозволу та статус. Така структура дозволяє створювати гнучкі дозволи для різних зон і строків дії.

Таблиці BUILDINGS, ACCESS_ZONES та ACCESS_DEVICES описують фізичну структуру об’єкта. Одна будівля може містити багато зон доступу, а кожна зона може мати один або декілька пристроїв доступу. Таблиця ACCESS_EVENTS фіксує результат кожної спроби входу або виходу. Вона пов’язується з цифровим ключем і пристроєм доступу.

Таблиця 3.2

Основні таблиці фізичної моделі бази даних

Таблиця	Призначення	Ключові поля
ROLES	Збереження ролей користувачів	role_id, role_name, role_description
USERS	Облік користувачів системи	user_id, role_id, full_name, email, phone, user_status
DIGITAL_KEYS	Збереження цифрових ключів і параметрів їх дії	key_id, user_id, key_code, activation_date, expiry_date, auth_method, key_status
ACCESS_PERMISSIONS	Збереження дозволів доступу цифрових ключів до зон	permission_id, key_id, zone_id, granted_by, granted_date, end_date, permission_status
BUILDINGS	Опис будівель, у яких використовується система	building_id, building_name, address, building_type
ACCESS_ZONES	Опис зон доступу в межах будівель	zone_id, building_id, zone_name, security_level, zone_status
ACCESS_DEVICES	Облік пристроїв зчитування ключів	device_id, zone_id, device_name, device_type, installation_point, device_status
ACCESS_EVENTS	Фіксація подій доступу, відмов і виходів	event_id, key_id, device_id, event_time, access_result, denial_reason, direction

Для підтримки цілісності даних у фізичній моделі передбачено зовнішні ключі між таблицями. ROLES пов’язана з USERS, USERS пов’язана з DIGITAL_KEYS, DIGITAL_KEYS пов’язана з ACCESS_PERMISSIONS та ACCESS_EVENTS, BUILDINGS пов’язана з ACCESS_ZONES, а ACCESS_ZONES пов’язана з ACCESS_DEVICES. Таке зв’язування дозволяє відстежувати, який користувач отримав ключ, до яких зон він має доступ і через який пристрій була здійснена спроба входу.

Фізична модель також підтримує аудит безпеки. Події доступу не видаляються разом із цифровим ключем або користувачем, а зберігаються як історичні записи. Завдяки цьому адміністратор або охоронець може переглянути історію спроб входу, визначити причини відмови та проаналізувати активність у конкретних зонах будівлі.

Таблиця 3.3

Основні зв’язки між таблицями бази даних

Зв’язок	Тип	Зміст
ROLES.role_id -> USERS.role_id	1:N	Одна роль може бути призначена багатьом користувачам
USERS.user_id -> DIGITAL_KEYS.user_id	1:N	Один користувач може мати декілька цифрових ключів
DIGITAL_KEYS.key_id -> ACCESS_PERMISSIONS.key_id	1:N	Один ключ може мати дозволи на декілька зон доступу
BUILDINGS.building_id -> ACCESS_ZONES.building_id	1:N	Одна будівля може містити декілька зон доступу
ACCESS_ZONES.zone_id -> ACCESS_DEVICES.zone_id	1:N	Одна зона може обслуговуватися декількома пристроями
ACCESS_DEVICES.device_id -> ACCESS_EVENTS.device_id	1:N	Один пристрій може реєструвати багато подій доступу

Розроблена фізична модель бази даних є достатньою для реалізації основних функцій системи керування доступом. Вона забезпечує збереження облікових даних, цифрових ключів, дозволів, зон будівлі, пристроїв доступу та подій безпеки. Структура не перевантажена зайвими сутностями, але дозволяє

надалі додати модулі розкладів доступу, гостьових ключів, біометричної перевірки або інтеграції з відеоспостереженням.

3.3 Алгоритмізація роботи програмних модулів системи

Алгоритмізація програмних модулів виконана для основних процесів системи: авторизації користувача, видачі цифрового ключа, використання ключа для входу, блокування або відкликання ключа та журналювання подій доступу. Саме ці процеси визначають логіку функціонування системи та забезпечують безпечний контроль доступу до будівель.

Першим базовим процесом є авторизація користувача. Вона передбачає введення логіна і пароля, отримання облікових даних із бази даних, перевірку їх правильності та надання доступу до системи. Якщо введені дані неправильні, користувач отримує повідомлення про помилку входу. Алгоритм авторизації наведено на рис. 3.2.

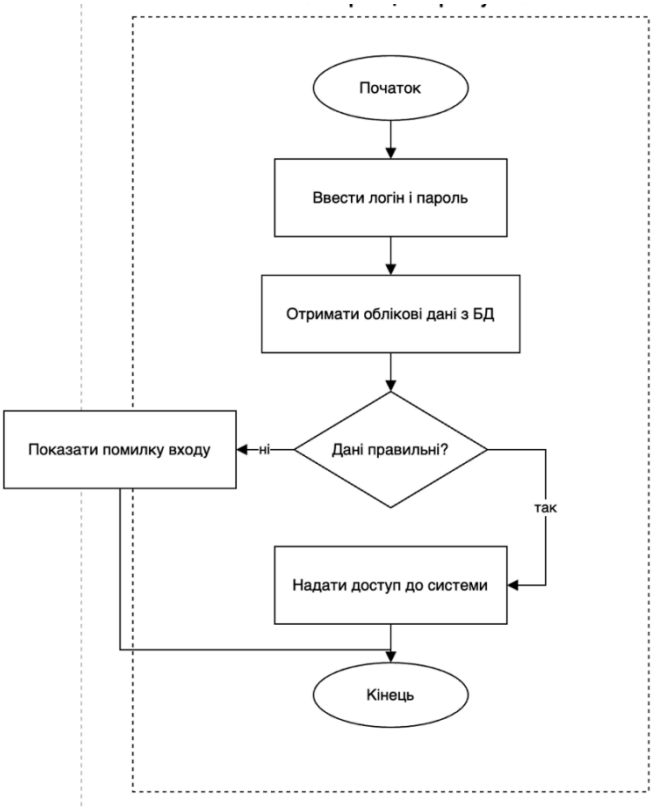

Рис. 3.2. Блок-схема алгоритму авторизації користувача

Алгоритм авторизації забезпечує первинний контроль доступу до програмного забезпечення. Він використовується для адміністратора, охоронця та інших користувачів, які мають працювати з системою. У разі успішної перевірки користувачу відкривається доступ до функцій відповідно до його ролі. У разі помилки система не надає доступу до модулів керування ключами й журналом подій.

Другим важливим процесом є видача цифрового ключа. Цей алгоритм використовується адміністратором або уповноваженою особою для створення ключа користувачу. Під час видачі ключа потрібно вибрати користувача, вказати параметри ключа, перевірити строки дії та права доступу. Якщо дані некоректні, система повертає оператора до редагування. Якщо параметри правильні, створюється новий цифровий ключ. Блок-схему алгоритму наведено на рис. 3.3.

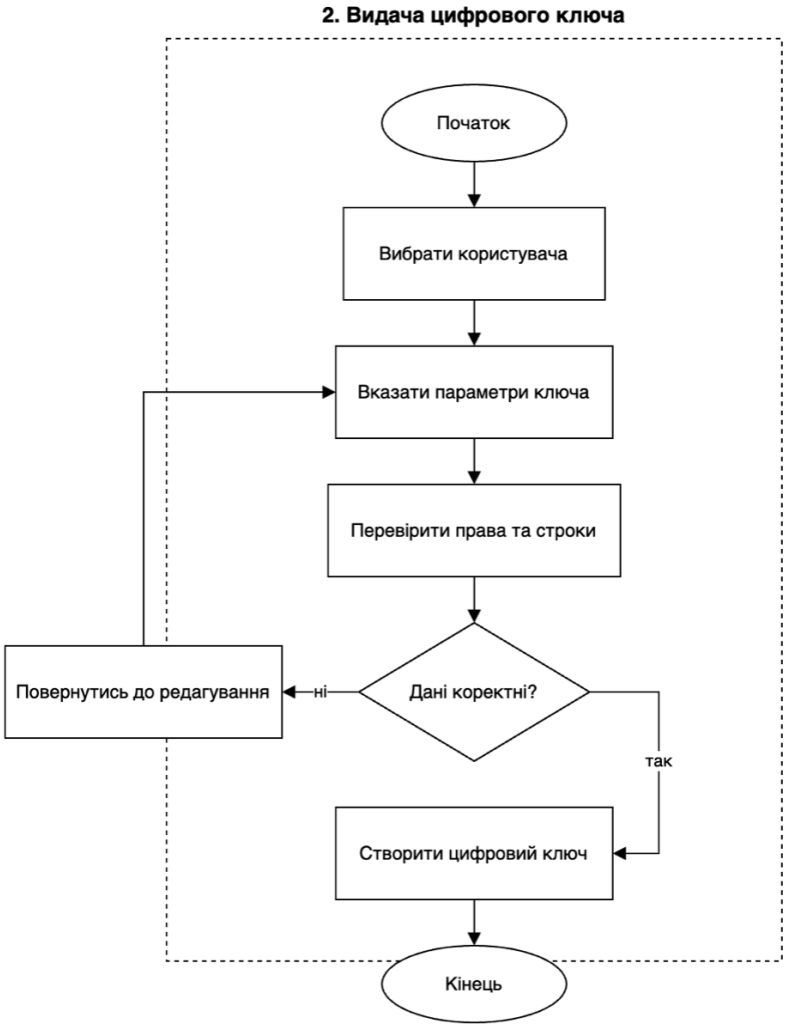

Рис. 3.3. Блок-схема алгоритму видачі цифрового ключа

У процесі видачі ключа особливе значення має перевірка строку дії. Ключ не повинен створюватися з некоректною датою завершення або без прив’язки до користувача. Крім того, цифровий ключ має отримати початковий статус, наприклад active або inactive, залежно від обраного сценарію. Це дозволяє уникнути ситуацій, коли ключ створено, але він не має коректних параметрів використання.

Третім процесом є використання цифрового ключа для входу. Користувач прикладає або надсилає ключ до пристрою доступу. Пристрій отримує дані ключа, після чого система перевіряє його активність. Якщо ключ неактивний, відкликаний або прострочений, доступ відхиляється. Якщо ключ активний, система перевіряє дозвіл для конкретної зони. Лише після підтвердження дозволу двері відкриваються. Алгоритм наведено на рис. 3.4.

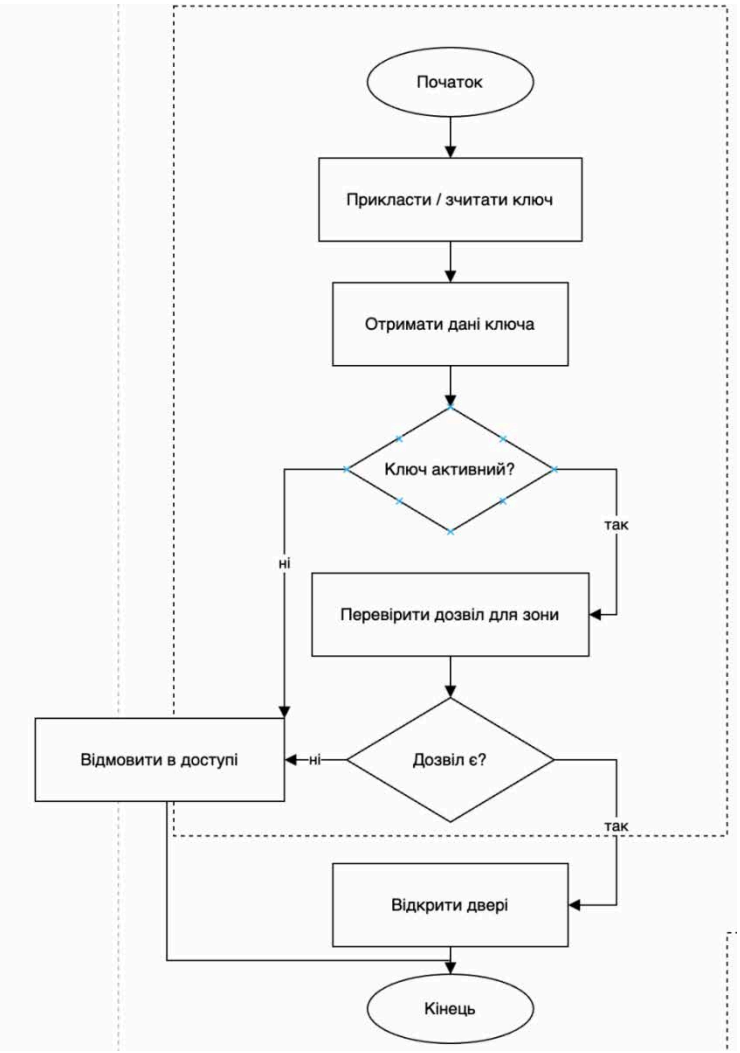

Рис. 3.4. Блок-схема алгоритму використання цифрового ключа для входу

Алгоритм використання ключа поєднує дві перевірки: стан самого ключа та наявність дозволу для конкретної зони. Така подвійна перевірка підвищує безпеку системи. Наприклад, ключ може бути активним, але не мати права входу до серверної кімнати або службового приміщення. У такому випадку пристрій доступу має відмовити у вході й передати подію до журналу.

Четвертим процесом є блокування або відкликання цифрового ключа. Цей алгоритм застосовується у випадках втрати смартфона, завершення строку доступу, звільнення працівника або виявлення підозрілої активності. Адміністратор вибирає цифровий ключ, обирає дію, підтверджує операцію, після чого система оновлює статус ключа. Якщо підтвердження не виконано, операція скасовується. Блок-схему наведено на рис. 3.5.

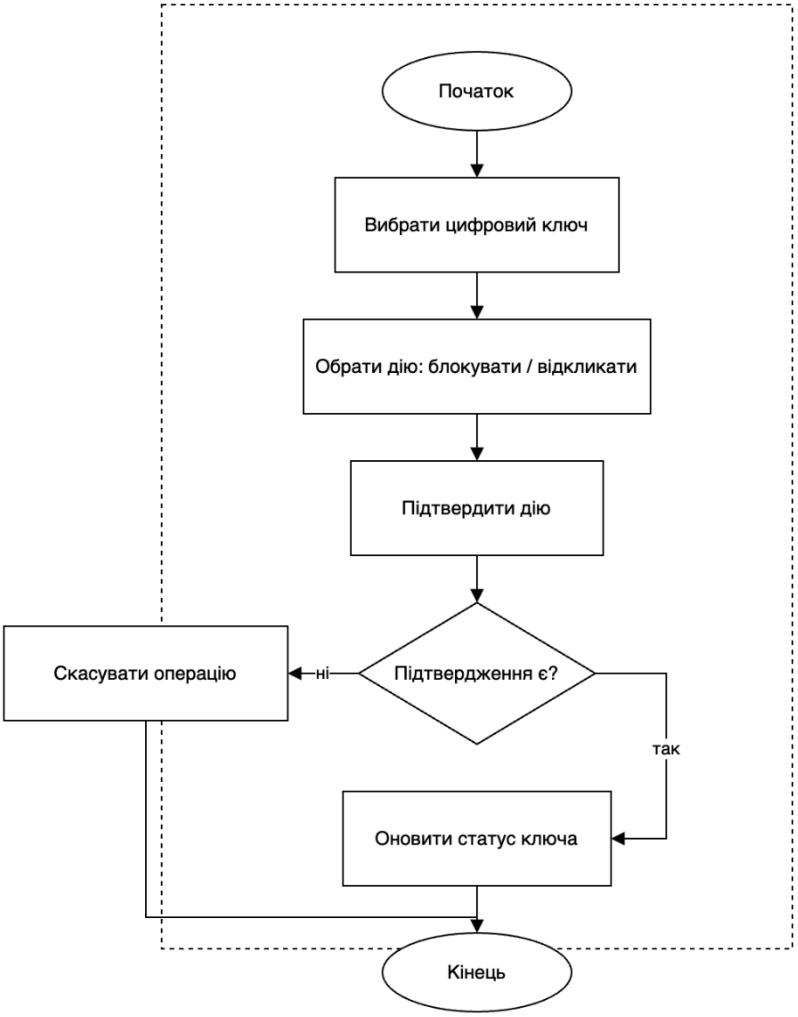

Рис. 3.5. Блок-схема алгоритму блокування або відкликання цифрового ключа

Блокування ключа є оборотною операцією, якщо адміністратор планує тимчасово заборонити доступ. Відкликання ключа використовується як остаточне припинення його дії. Обидві операції мають фіксуватися в журналі подій, оскільки зміна статусу ключа безпосередньо впливає на безпеку будівлі.

П’ятим процесом є журналювання подій доступу. Після кожної спроби входу пристрій передає до системи інформацію про ключ, зону, результат перевірки та час події. Система формує запис події, зберігає його в базі даних і, за потреби, надсилає сповіщення адміністратору. Блок-схему журналювання наведено на рис. 3.6.

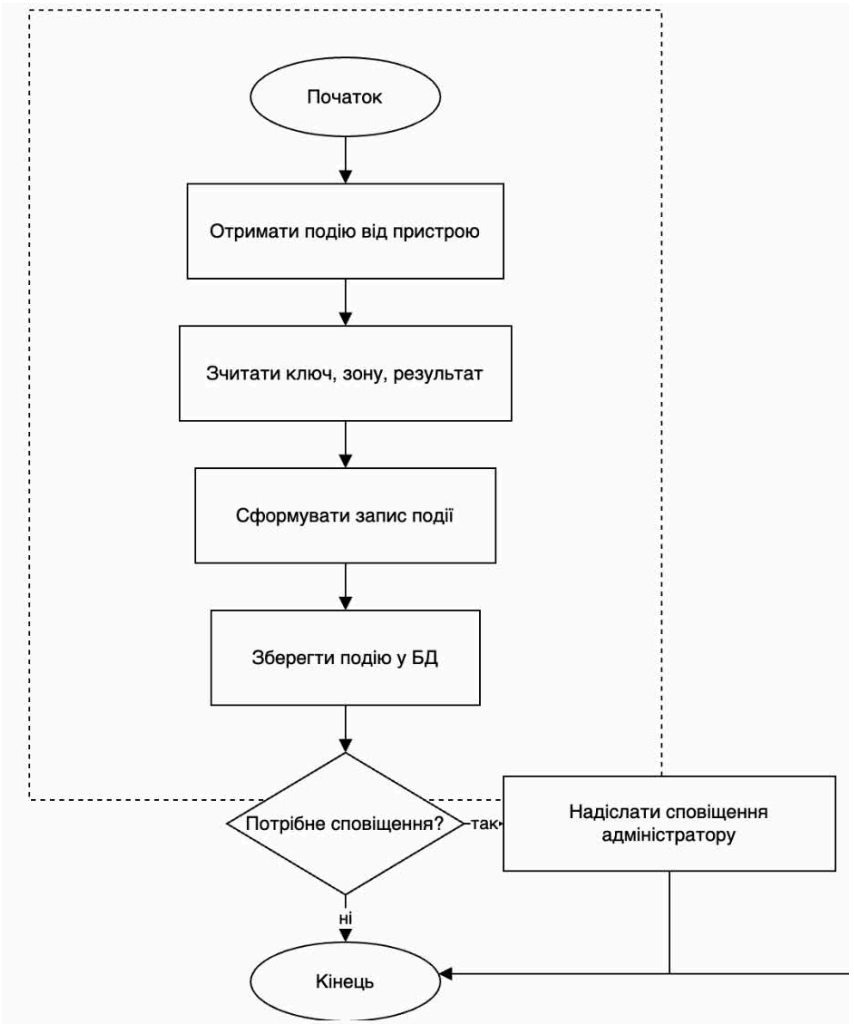

Рис. 3.6. Блок-схема алгоритму журналювання подій доступу

Журналювання є обов’язковим для систем керування доступом, оскільки воно забезпечує можливість аналізу подій безпеки. До журналу записуються як успішні входи, так і відмови. Якщо подія має підвищений ризик, наприклад багаторазова відмова доступу або спроба використання відкликаного ключа, система може сформувати сповіщення для адміністратора або охоронця.

Узагальнену характеристику основних алгоритмів системи наведено у таблиці 3.4.

Таблиця 3.4

Основні алгоритми програмного забезпечення

Алгоритм	Вхідні дані	Результат
Авторизація користувача	Логін, пароль	Надання доступу до системи або повідомлення про помилку

Видача цифрового ключа	Користувач, параметри ключа, строки дії	Створення цифрового ключа в базі даних
Використання цифрового ключа	Код ключа, зона, пристрій доступу	Дозвіл або заборона входу
Блокування / відкликання ключа	Ключ, тип операції, підтвердження адміністратора	Оновлення статусу ключа
Журналювання події	Подія від пристрою, ключ, зона, результат	Запис події в базу даних і можливе сповіщення

Розроблені алгоритми забезпечують повний цикл роботи системи з цифровими ключами. Вони враховують не лише штатні операції, а й помилкові або небезпечні ситуації: некоректні дані авторизації, неактивний ключ, відсутність дозволу для зони, скасування операції відкликання та потребу в сповіщенні адміністратора про подію безпеки.

3.4 Реалізація основних програмних модулів

Програмне забезпечення системи керування доступом реалізується як набір функціональних модулів, кожен з яких відповідає за окрему частину бізнес-логіки. Модульна структура спрощує розробку, тестування та подальший супровід системи. Основними модулями є модуль автентифікації, модуль користувачів і ролей, модуль цифрових ключів, модуль прав доступу, модуль зон і пристроїв, модуль журналу подій та модуль сповіщень.

Модуль автентифікації забезпечує перевірку облікових даних і створення сеансу роботи користувача. Для адміністратора він відкриває доступ до керування всіма сутностями системи. Для охоронця доступ обмежується переглядом подій, контролем запитів та оперативним реагуванням. Для звичайного користувача основними функціями є подання запиту, отримання ключа та використання його для входу.

Модуль користувачів і ролей відповідає за створення, редагування та блокування облікових записів. Під час створення користувача задається роль,

контактна інформація та статус. Використання ролей дозволяє відокремити службові можливості адміністратора, охоронця та користувача. Це спрощує контроль прав доступу до функцій системи.

Модуль цифрових ключів реалізує життєвий цикл ключа. Він створює ключ, призначає його користувачу, зберігає строк дії, метод автентифікації та статус. У разі потреби модуль змінює статус ключа на blocked, revoked або expired. Це дозволяє швидко припинити доступ користувача без зміни фізичної інфраструктури будівлі.

Модуль прав доступу визначає, до яких зон будівлі може застосовуватися конкретний цифровий ключ. Він перевіряє, чи активний дозвіл, чи не завершився його строк дії та чи відповідає зона рівню доступу користувача. Саме цей модуль приймає рішення про дозвіл або заборону входу після зчитування ключа пристроєм доступу.

Модуль зон і пристроїв використовується для опису будівель, зон доступу та пристроїв зчитування. Для кожної зони визначається рівень безпеки та поточний статус. Пристрої доступу прив’язуються до конкретних зон і передають події до серверної частини. Це дозволяє адміністратору контролювати технічну структуру об’єкта.

Модуль журналу подій забезпечує збереження всіх спроб доступу. У журналі фіксується час події, ключ, пристрій, результат, напрям руху та причина відмови. Завдяки цьому можна відстежувати активність користувачів, аналізувати небезпечні ситуації та готувати звіти для служби безпеки.

Таблиця 3.5

Основні програмні модулі системи

Модуль	Призначення
Модуль автентифікації	Перевірка користувача, ролі та права входу до системи
Модуль користувачів і ролей	Створення, редагування та блокування користувачів
Модуль цифрових ключів	Видача, активація, блокування та відкликання цифрових ключів

Модуль прав доступу	Призначення і перевірка дозволів для зон будівлі
Модуль будівель і зон	Опис будівель, зон доступу та рівнів безпеки
Модуль пристроїв доступу	Облік пристроїв зчитування ключів і їхнього стану
Модуль журналу подій	Фіксація успішних входів, відмов і службових подій
Модуль сповіщень	Передавання повідомлень адміністратору або охоронцю

Реалізація програмних модулів передбачає використання єдиного шару доступу до даних. Це означає, що модулі не звертаються до таблиць бази напряму, а використовують підготовлені функції або сервіси. Наприклад, модуль цифрових ключів викликає метод створення ключа, модуль прав доступу - метод перевірки дозволу, а модуль журналу - метод збереження події.

Такий підхід зменшує залежність бізнес-логіки від конкретної структури бази даних. Якщо надалі буде прийнято рішення перейти з SQLite на PostgreSQL або додати серверну реплікацію, більшість прикладних модулів можна буде залишити без істотних змін.

3.5 Опис інтерфейсу користувача

Інтерфейс системи керування доступом має забезпечувати зручну роботу для трьох основних ролей: адміністратора, охоронця та користувача. Для адміністратора важливими є функції створення користувачів, видачі ключів, призначення прав доступу, перегляду журналу та блокування ключів. Для охоронця головними є перегляд подій, підтвердження запитів і контроль спроб доступу. Для користувача інтерфейс має бути максимально простим: авторизація, отримання ключа, перегляд статусу ключа та використання його для входу.

Адміністративний інтерфейс доцільно організувати у вигляді веб-панелі з окремими розділами: користувачі, ролі, цифрові ключі, будівлі, зони, пристрої,

дозволи, журнал подій і сповіщення. На головній сторінці може відображатися кількість активних користувачів, активних ключів, останніх подій доступу, заблокованих ключів та відмов у доступі.

Мобільний інтерфейс користувача має містити мінімальний набір дій. Після входу користувач бачить доступні цифрові ключі, строк їх дії та статус. Під час входу до будівлі ключ може передаватися через NFC, QR-код, BLE або інший підтримуваний метод. У разі успішного входу користувач отримує повідомлення про підтвердження, а подія фіксується в журналі.

Інтерфейс охоронця має бути орієнтований на оперативну роботу. Він повинен відображати список останніх подій доступу, запити на погодження, відмови та спроби використання заблокованих ключів. Для кожної події необхідно показувати час, користувача, зону, пристрій, результат і причину відмови. Це дозволяє швидко реагувати на нестандартні ситуації.

Таблиця 3.6

Основні екрани інтерфейсу користувача

Екран	Користувач	Призначення
Вхід до системи	Усі ролі	Авторизація користувача за логіном і паролем
Панель адміністратора	Адміністратор	Перегляд загальних показників і швидкий доступ до модулів
Користувачі та ролі	Адміністратор	Керування обліковими записами та ролями
Цифрові ключі	Адміністратор	Видача, блокування й відкликання ключів
Зони та пристрої	Адміністратор	Облік будівель, зон і пристроїв доступу

Продовження таблиці 3.6

Журнал доступу	Адміністратор, охоронець	Перегляд спроб входу, відмов і подій безпеки
Мій цифровий ключ	Користувач	Перегляд активного ключа та його використання для входу

Важливою вимогою до інтерфейсу є зрозумілість статусів. Для цифрового ключа користувач повинен бачити, чи він активний, заблокований, відкликаний або прострочений. Для дозволу доступу потрібно відображати, до якої зони він належить і до якої дати діє. Для події доступу необхідно показувати результат: granted або denied, а також причину відмови.

Інтерфейс має не лише забезпечувати введення даних, а й мінімізувати кількість помилок користувача. Наприклад, під час видачі ключа система повинна перевіряти, чи вибрано користувача, чи задано дату завершення дії, чи не дублюється код ключа і чи існує потрібна зона доступу. Завдяки цьому підвищується надійність роботи системи та зменшується ймовірність некоректних дозволів.

3.6 Висновки до третього розділу

У третьому розділі виконано розроблення та опис програмної реалізації системи керування доступом до будівель з використанням цифрових ключів. Обґрунтовано вибір технологічних засобів, які забезпечують створення клієнт-серверної системи з веб-інтерфейсом адміністратора, мобільним клієнтом користувача, REST API та реляційною базою даних.

Розроблено фізичну модель бази даних AccessControlDB. Вона включає таблиці ролей, користувачів, цифрових ключів, прав доступу, будівель, зон, пристроїв і подій доступу. Визначені первинні та зовнішні ключі забезпечують цілісність даних, а структура таблиць підтримує повний цикл роботи системи: від створення користувача до журналювання фактичної події входу.

Описано алгоритми основних програмних модулів: авторизації користувача, видачі цифрового ключа, використання ключа для входу, блокування або відкликання ключа та журналювання подій доступу. Алгоритми враховують як штатні сценарії, так і ситуації відмови, зокрема неправильні

облікові дані, неактивний ключ, відсутність дозволу або скасування операції адміністратором.

Також розглянуто реалізацію основних програмних модулів і структуру інтерфейсу користувача. Запропонована система є модульною, підтримує розмежування ролей, забезпечує аудит подій та може бути розширена в майбутньому шляхом додавання гостьових ключів, біометричної автентифікації, інтеграції з відеоспостереженням або хмарним сховищем подій.

4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 План тестування програмних модулів та методика оцінювання результатів

Тестування системи керування доступом до будівель з використанням цифрових ключів виконувалося з метою перевірки працездатності основних програмних модулів, коректності оброблення цифрових ключів, надійності автентифікації користувачів, правильності фіксації подій доступу та стабільності взаємодії між клієнтською частиною, сервером застосунку, терміналом доступу і базою даних.

Особливість цієї системи полягає в тому, що вона поєднує програмні засоби адміністрування, мобільний доступ користувача, серверну обробку прав доступу та роботу пристроїв зчитування QR/NFC-ключів. Тому тестування охоплює не лише окремі екрани інтерфейсу, а й наскрізні сценарії: авторизацію адміністратора, створення цифрового ключа, перевірку дозволу для зони, реєстрацію події проходу, відмову в доступі та формування критичного сповіщення.

Методика тестування базувалася на порівнянні фактичної поведінки системи з очікуваними результатами. Для кожного модуля визначено вхідні дані, очікуваний результат і критерії приймання. Основний план тестування подано у таблиці 4.1.

Таблиця 4.1

План тестування програмних модулів системи керування доступом

№	Модуль / функція	Вхідні дані	Очікуваний результат	Критерії приймання
1	Авторизація користувача	Електронна пошта, пароль, одноразовий код	Вхід до системи або повідомлення про помилку	Доступ надається лише за коректними даними

Продовження таблиці 4.1

2	Керування ролями	Роль адміністратора, охоронця або користувача	Відкриття лише дозволених функцій	Функції відповідають ролі користувача
3	Створення цифрового ключа	Користувач, роль, тип ключа, строк дії, дозволені зони	Формування активного ключа	Ключ зберігається і має правильні параметри
4	Перевірка цифрового ключа	QR/NFC-ключ на терміналі доступу	Дозвіл або відмова у вході	Рішення відповідає правам доступу
5	Керування зонами доступу	Навчальний корпус, лабораторія, серверна кімната	Призначення рівня доступу для зони	Зона враховується при перевірці ключа
6	Блокування або відкликання ключа	Вибраний цифровий ключ і підтвердження дії	Зміна статусу ключа	Заблокований ключ не відкриває двері
7	Журнал подій	Події входу, відмови, перевірки та блокування	Запис подій у базу даних	Журнал містить час, користувача, зону, пристрій і результат
8	Критичні сповіщення	Повторні відмови, прострочений ключ, offline-пристрій	Відображення повідомлення адміністратору	Подія потрапляє до блоку критичних сповіщень
9	Експлуатаційна схема	Клієнт, сервер, термінал доступу, база даних	Передавання даних через HTTPS/TLS, REST API і SQL	Компоненти взаємодіють без втрати подій

Оцінювання результатів виконувалося за функціональними, безпековими та експлуатаційними критеріями. До функціональних критеріїв належать правильність входу в систему, створення ключа, призначення дозволених зон і запис подій доступу. До безпекових критеріїв належать перевірка ролей, строків дії ключів, блокування відкликаних ключів і журналювання кожної спроби входу. До експлуатаційних критеріїв належать зрозумілість інтерфейсу,

швидкість переходу між модулями та стабільність взаємодії серверної частини з терміналом доступу.

Тестування проводилося для основних ролей системи: адміністратора, охоронця та користувача. Адміністратор перевіряв створення ключів, управління правами та перегляд журналу. Охоронець контролював останні події доступу та критичні сповіщення. Користувач використовував цифровий ключ для входу до дозволених зон. Такий підхід дозволяє перевірити систему не лише за окремими функціями, а й у реальних сценаріях експлуатації.

4.2 Тестування системи керування доступом до будівель з використанням цифрових ключів

Тестування системи керування доступом до будівель з використанням цифрових ключів розпочато з перевірки модуля авторизації. Вікно входу призначене для перевірки особи користувача та визначення його ролі в системі. На формі передбачено вибір ролі адміністратора, охоронця або користувача, введення електронної адреси, пароля, типу автентифікації та коду підтвердження. Наявність одноразового коду підвищує рівень захисту, оскільки для входу недостатньо лише знати пароль.

Під час тестування перевірено сценарій входу адміністратора з використанням електронної адреси admin@university.edu.ua, пароля та коду підтвердження. Після успішної перевірки система переходить до робочого інтерфейсу. У разі неправильного пароля або коду підтвердження доступ не надається. Результат перевірки форми авторизації наведено на рисунку 4.1.

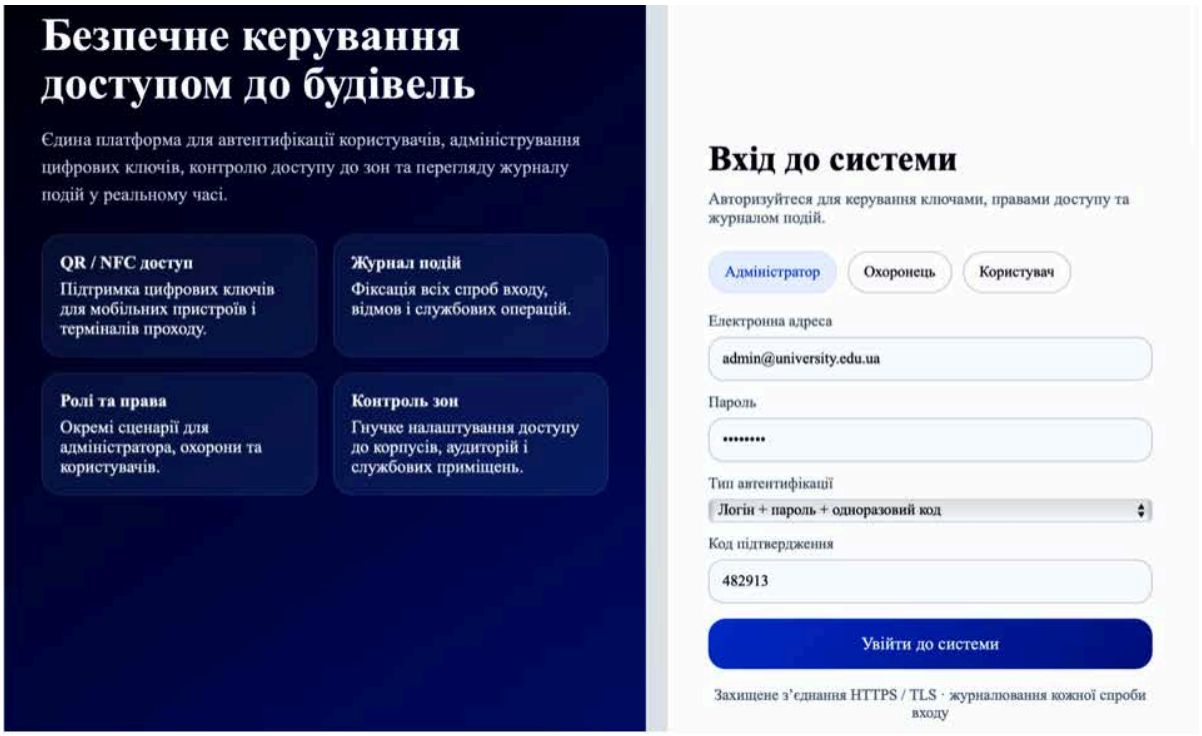

Рис. 4.1. Форма авторизації користувача системи

Після авторизації було перевірено панель оперативного моніторингу. Цей модуль призначений для роботи охоронця або адміністратора, оскільки на ньому відображаються останні події доступу, критичні сповіщення і стан зон доступу. У таблиці останніх подій показано час, користувача, зону, пристрій і результат спроби входу. Система відрізняє успішний доступ, відмову та ситуації, що потребують додаткової перевірки.

На панелі критичних сповіщень відображаються події, які потребують оперативної реакції: кілька послідовних відмов входу на одному терміналі, завершення строку дії цифрового ключа або втрата зв'язку з пристроєм. Окремий блок показує стан зон доступу за сегментами, зокрема навчальні корпуси, лабораторії та службові приміщення. Результат перевірки панелі моніторингу наведено на рисунку 4.2.

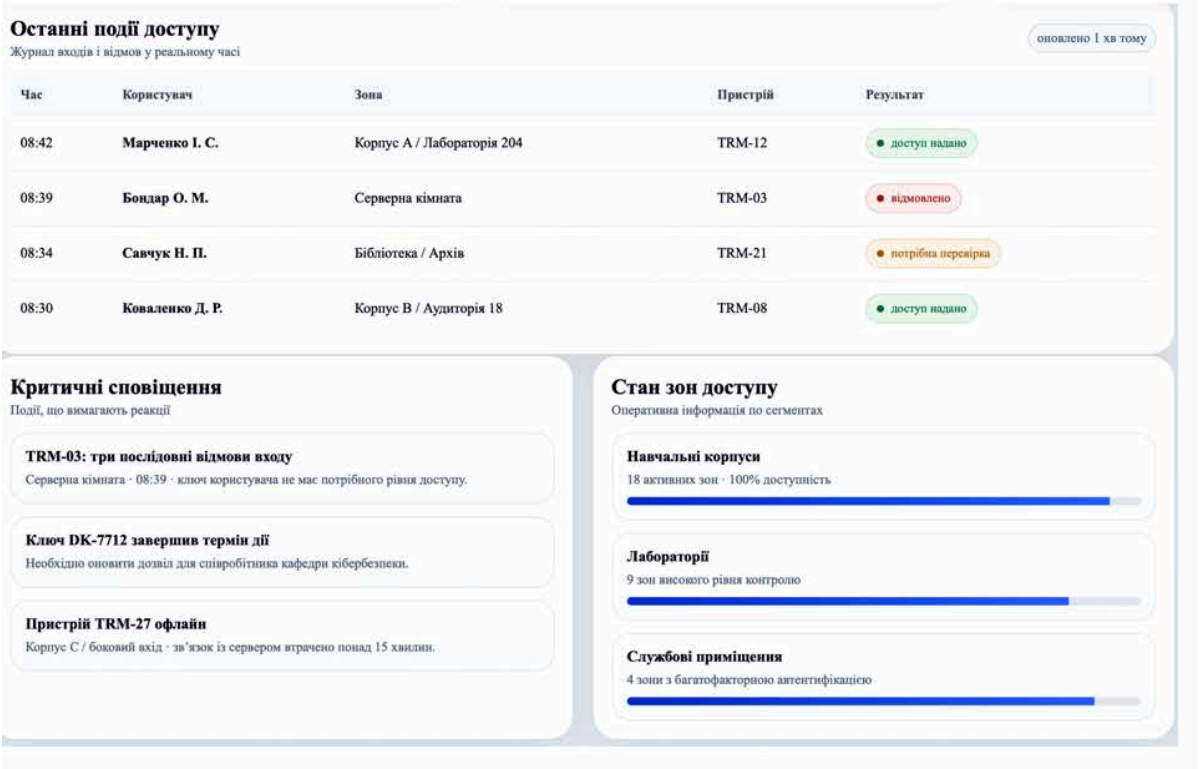

Рис. 4.2. Панель моніторингу подій доступу та стану зон

Наступним етапом протестовано модуль швидкого створення цифрового ключа. Він використовується адміністратором для видачі нового цифрового ключа конкретному користувачу. У формі зазначаються дані користувача, його роль, тип ключа, строк дії, перелік дозволених зон і службова примітка. У тестовому сценарії для користувача Пилипчук Анни Олексіївни було обрано роль «Викладач», тип ключа «Mobile App + NFC», строк дії до 31.12.2026 та дозволені зони: корпус А, аудиторії 201–214 і бібліотека.

Під час перевірки встановлено, що форма дозволяє структурувати параметри ключа до його генерації. Це зменшує ризик помилкового надання доступу, оскільки адміністратор бачить роль користувача, строк чинності й перелік дозволених зон до натискання кнопки генерації. Інтерфейс модуля швидкого створення ключа наведено на рисунку 4.3.

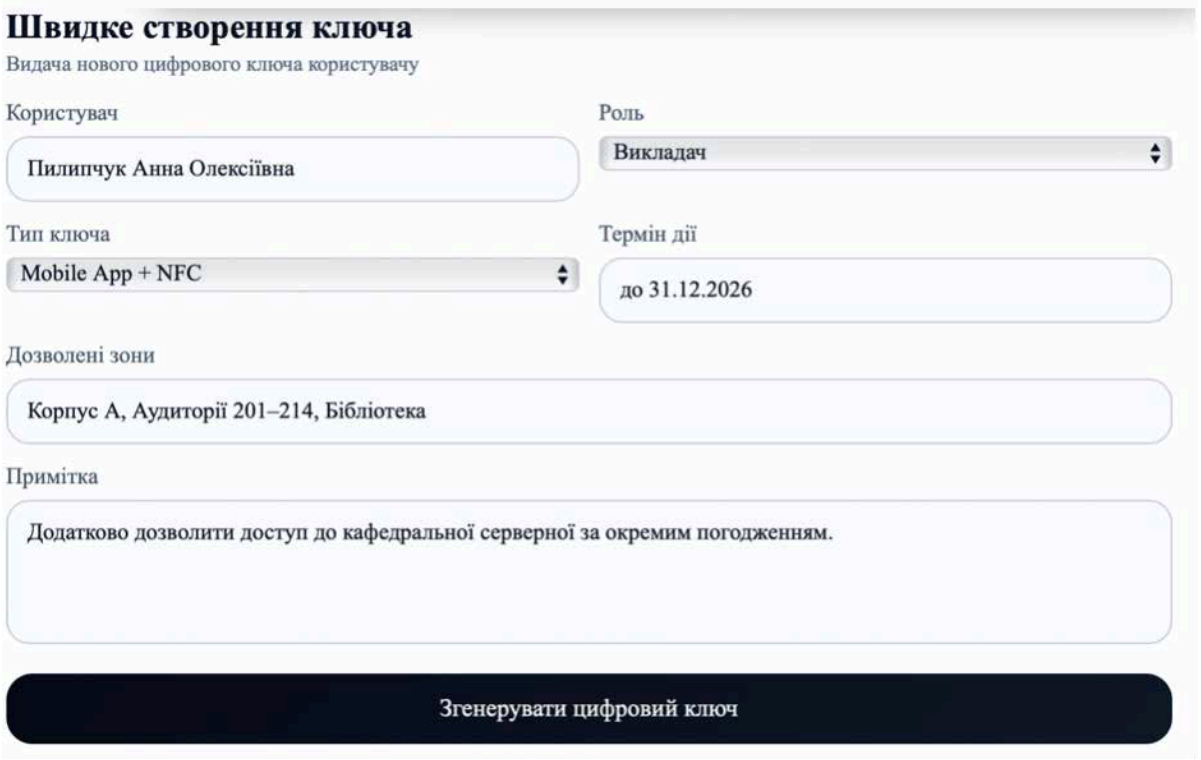

Рис. 4.3. Форма швидкого створення цифрового ключа

Окремо перевірено логіку використання цифрового ключа на терміналі доступу. Після прикладання NFC-ключа або зчитування QR-коду термінал передає дані на сервер застосунку. Далі система визначає користувача, перевіряє активність ключа, строк його дії, дозволену зону та стан пристрою. Якщо ключ активний і користувач має дозвіл на обрану зону, система дозволяє вхід і створює запис у журналі. Якщо ключ відкликаний, прострочений або не має дозволу для зони, формується відмова.

Під час тестування було перевірено три основні варіанти результату: доступ надано, доступ відмовлено та потрібна додаткова перевірка. У першому випадку подія фіксується як granted, у другому як denied із зазначенням причини відмови, у третьому як подія контролю, що може вимагати дії охоронця. Такий підхід дозволяє зберігати повну історію проходів і спроб несанкціонованого входу.

Таблиця 4.2

Результати тестування основних модулів системи

Модуль	Перевірена дія	Очікуваний результат	Результат
Авторизація	Вхід адміністратора з паролем і кодом підтвердження	Відкриття робочого інтерфейсу	Виконано
Авторизація	Введення неправильного пароля або коду	Повідомлення про помилку входу	Виконано
Моніторинг подій	Перегляд останніх подій доступу	Відображення часу, користувача, зони, пристрою та результату	Виконано
Сповіщення	Перевірка критичних повідомлень	Відображення повторних відмов, завершення строку ключа, offline-пристрою	Виконано
Створення ключа	Заповнення форми нового ключа	Формування ключа з роллю, строком дії та дозволеними зонами	Виконано
Перевірка ключа	Зчитування QR/NFC-ключа терміналом	Дозвіл або відмова залежно від прав доступу	Виконано
Права доступу	Перевірка доступу до різних зон	Доступ дозволяється лише до визначених зон	Виконано
Журнал подій	Фіксація спроб входу	Події зберігаються з результатом і причиною відмови	Виконано
Пристрої доступу	Перевірка стану термінала	Система відображає активність або втрату зв’язку з пристроєм	Виконано

Результати тестування свідчать, що основні функціональні модулі працюють коректно. Система забезпечує контрольований вхід користувачів, формування цифрових ключів, перевірку прав доступу, фіксацію подій і відображення критичних сповіщень. Найважливішим результатом є те, що перевірка ключа виконується не лише за фактом його наявності, а й за статусом, строком дії, роллю користувача та дозволеною зоною.

Для перевірки експлуатаційної структури системи розглянуто схему розгортання. Вона включає робоче місце адміністратора, мобільний пристрій користувача, термінал доступу, сервер застосунку та сервер бази даних. Адміністратор і мобільний клієнт взаємодіють із сервером через HTTPS/TCP/IP, термінал доступу передає події через REST API/TLS, а сервер застосунку звертається до бази даних через TCP/IP і SQL. Діаграма розгортання подана на рисунку 4.4.

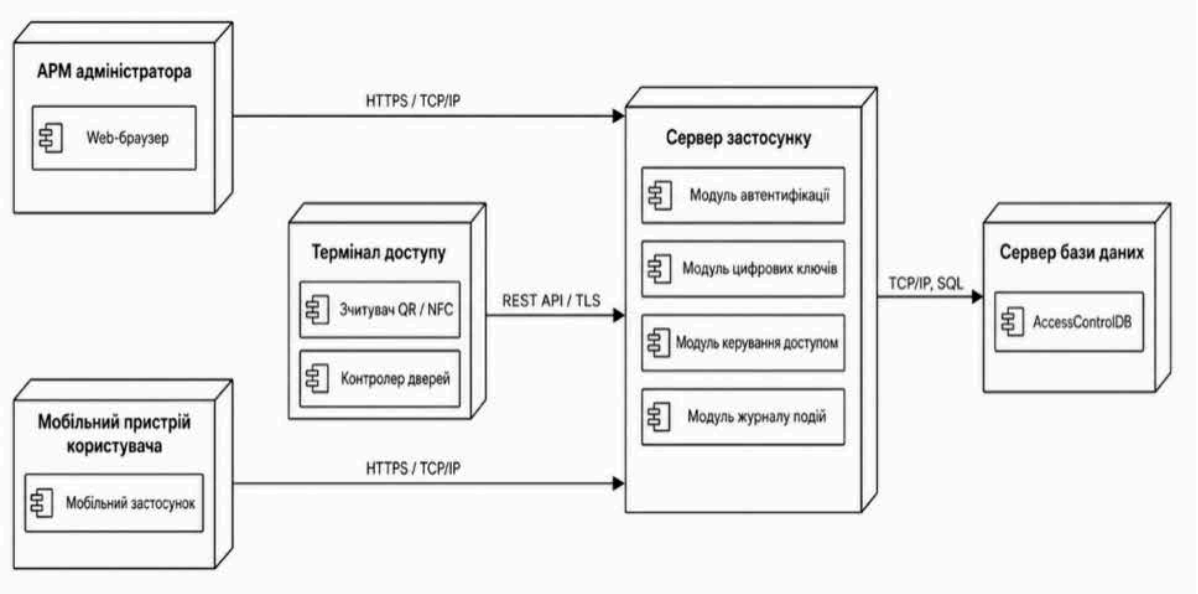

Рис. 4.4. Діаграма розгортання системи керування доступом до будівель

Перевірка схеми розгортання підтвердила логічну узгодженість компонентів. Модуль автентифікації відповідає за вхід користувачів, модуль цифрових ключів за створення та перевірку ключів, модуль керування доступом за прийняття рішення щодо проходу, а модуль журналу подій за збереження інформації про кожну спробу входу. Сервер бази даних AccessControlDB зберігає користувачів, ролі, ключі, дозволи, зони, пристрої та події доступу.

Таким чином, тестування підтвердило працездатність системи у ключових сценаріях експлуатації: адміністрування цифрових ключів, доступ користувача до будівлі, контроль подій охоронцем і фіксація службової інформації в журналі. Отримані результати свідчать про готовність програмного

забезпечення до подальшого розширення та використання як прототипу системи керування доступом до будівель.

4.3 Оцінювання ефективності роботи системи

Оцінювання ефективності системи керування доступом до будівель з використанням цифрових ключів виконувалося за функціональними, безпековими та організаційними критеріями. Функціональна ефективність визначається здатністю системи забезпечувати повний цикл роботи з цифровими ключами: створення, видачу, перевірку, блокування, журналювання та аналіз подій доступу. Безпекова ефективність пов’язана з контролем ролей, строків дії ключів, відмов у доступі та сповіщень про критичні події. Організаційна ефективність полягає у зменшенні залежності від фізичних ключів і паперових журналів.

У традиційних системах доступу значна частина операцій виконується вручну: ключі або картки видаються фізично, інформація про доступ часто зберігається в окремих документах, а блокування втраченого ключа потребує додаткового втручання. Розроблена система переносить ці процеси у цифрове середовище, що дає змогу швидко змінювати права доступу, відкликати ключі, переглядати журнал подій і контролювати стан зон у реальному часі.

Таблиця 4.3

Порівняння традиційного підходу та розробленої системи

Критерій	Традиційний підхід	Розроблена система
Видача доступу	Фізична передача ключа або картки	Створення цифрового ключа через інтерфейс адміністратора
Зміна прав доступу	Потребує ручного оновлення або заміни носія	Права змінюються в системі без фізичної заміни ключа
Блокування ключа	Може потребувати вилучення ключа або заміни замка	Ключ блокується або відкликається програмно
Контроль подій	Журнал ведеться вручну або частково автоматизовано	Кожна подія фіксується в електронному журналі

Реакція на інциденти	Залежить від повідомлення охорони	Система формує критичні сповіщення
Масштабування	Ускладнюється при збільшенні кількості користувачів	Додавання користувачів, зон і пристроїв виконується через БД

Порівняння показує, що основна перевага розробленої системи полягає у централізованому керуванні доступом. Адміністратор може створити ключ, визначити строк його дії, призначити дозволені зони та за потреби відкликати доступ без фізичного втручання у дверне обладнання. Це особливо важливо для будівель із багатьма приміщеннями, різними групами користувачів і змінними правами доступу.

Ефективність системи також проявляється у підвищенні прозорості подій. Охоронець або адміністратор бачить останні спроби входу, причини відмов, активні інциденти та стан зон. Наявність електронного журналу дозволяє швидко встановити, хто, коли і через який пристрій намагався отримати доступ. Це підвищує контрольованість процесу та створює основу для службового аудиту.

Для узагальнення результатів оцінювання сформовано перелік показників ефективності, наведений у таблиці 4.4.

Таблиця 4.4

Показники ефективності програмного забезпечення

Показник	Реалізація в системі	Очікуваний ефект
Час створення ключа	Ключ формується через одну форму з вибором користувача, ролі, типу та зон	Скорочення часу адміністрування
Контроль строку дії	Для кожного ключа задається дата завершення	Зменшення ризику використання прострочених ключів
Безпека доступу	Перевіряються роль, зона, статус ключа та строк дії	Зменшення кількості несанкціонованих входів
Журналювання	Події записуються з часом, користувачем, зоною, пристроєм і результатом	Підвищення прозорості та контрольованості
Реакція на інциденти	Критичні події виводяться на панель сповіщень	Швидше реагування охорони або адміністратора

Масштабованість	Структура БД підтримує користувачів, ключі, пристрої, будівлі та зони	Можливість розширення системи

З наведених показників видно, що розроблена система підвищує ефективність керування доступом за рахунок автоматизації типових операцій і централізованого контролю. Основний практичний ефект полягає у скороченні часу на видачу та зміну ключів, підвищенні безпеки при роботі зі строками дії, а також у можливості швидкого аналізу подій доступу.

Водночас система має потенціал для подальшого розвитку. У перспективі її можна доповнити підтримкою біометричної автентифікації, інтеграцією з корпоративним каталогом користувачів, мобільними push-сповіщеннями, резервним сервером, модулем аналітики ризикових подій і веб-панеллю для керівника служби безпеки. Це дозволить перетворити прототип на повноцінну систему контролю доступу для навчальних корпусів, офісних будівель або виробничих об’єктів.

4.4 Висновки до четвертого розділу

У четвертому розділі проведено тестування та оцінювання ефективності системи керування доступом до будівель з використанням цифрових ключів. Розроблено план тестування, який охоплює авторизацію користувачів, створення цифрових ключів, перевірку прав доступу, блокування ключів, журналювання подій, формування критичних сповіщень і перевірку структури розгортання системи.

Під час тестування встановлено, що система коректно виконує основні сценарії роботи. Авторизація забезпечує контрольований вхід до системи, модуль створення ключів дозволяє призначати користувачам тип ключа, строк дії та дозволені зони, а модуль моніторингу відображає останні події доступу, відмови та критичні сповіщення. Перевірка використання ключа підтвердила,

що рішення про доступ приймається з урахуванням активності ключа, прав користувача і зони проходу.

Оцінювання ефективності показало, що розроблена система має переваги порівняно з традиційним підходом до керування доступом. Вона спрощує видачу ключів, дозволяє швидко змінювати або відкликати права доступу, забезпечує електронне журналювання подій і підвищує оперативність реагування на інциденти. Наявність централізованої бази даних і модульної структури створює можливість подальшого масштабування системи.

Отже, результати тестування підтвердили працездатність програмного забезпечення та відповідність реалізованих модулів поставленим вимогам. Система може використовуватися як програмний прототип для організації цифрового доступу до будівель, контролю подій входу та підтримки службового аудиту безпеки.

ВИСНОВКИ

У бакалаврській кваліфікаційній роботі розроблено програмне забезпечення системи керування доступом до будівель з використанням цифрових ключів. Запропоноване рішення орієнтоване на автоматизацію процесів автентифікації користувачів, видачі цифрових ключів, призначення прав доступу, перевірки дозволів у зонах будівлі та фіксації подій входу або відмови в доступі.

У першому розділі виконано аналіз предметної області. Визначено, що сучасні системи доступу переходять від фізичних ключів і пластикових карток до мобільних і цифрових ідентифікаторів, які зручніше адмініструвати, обмежувати за строком дії та швидко відкликати у разі зміни статусу користувача. Розглянуто існуючі рішення, зокрема HID Mobile Access, SALTO KS, Kisi, Brivo Mobile Pass та Avigilon Alta Access, що дало змогу визначити типові функції таких систем і сформувати вимоги до розроблюваного програмного забезпечення.

У другому розділі здійснено проєктування інформаційного та програмного забезпечення. Розроблено логічну модель даних, яка охоплює ролі, користувачів, цифрові ключі, права доступу, будівлі, зони доступу, пристрої та події доступу. Побудовано діаграму класів, кооперації, діаграму компонентів і діаграму пакетів. Це дозволило формалізувати структуру системи та визначити взаємодію між основними модулями.

У третьому розділі описано реалізацію програмного забезпечення. Обґрунтовано вибір технологічних засобів, розроблено фізичну модель бази даних AccessControlDB, визначено структуру таблиць і ключових зв’язків. Також подано алгоритми авторизації, видачі цифрового ключа, перевірки доступу, блокування або відкликання ключа та журналювання подій. Описано основні програмні модулі та інтерфейс користувача.

У четвертому розділі проведено тестування та оцінювання ефективності системи. Перевірено форму входу, панель моніторингу подій, швидке створення

цифрового ключа, роботу журналу подій, оброблення критичних сповіщень і загальну схему розгортання. Результати тестування підтвердили, що система коректно виконує основні функції, забезпечує фіксацію подій доступу та дає змогу адміністратору оперативно керувати ключами й правами користувачів.

Розроблене програмне забезпечення підвищує зручність адміністрування доступу, зменшує залежність від фізичних носіїв, скорочує час видачі та відкликання ключів і забезпечує прозорий журнал подій. У подальшому систему можна розширити за рахунок інтеграції з реальними NFC/QR-терміналами, підтримки біометричної автентифікації, інтеграції з Active Directory або Google Workspace, розширеного модуля аналітики ризикових подій і мобільного застосунку для користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Київ : ДП «УкрНДНЦ», 2016.

2. ДСТУ ISO/IEC 27001:2023. Інформаційна безпека, кібербезпека та захист конфіденційності. Системи керування інформаційною безпекою. Вимоги. Київ : ДП «УкрНДНЦ», 2023.

3. ДСТУ ISO/IEC 27002:2023. Інформаційна безпека, кібербезпека та захист конфіденційності. Заходи керування інформаційною безпекою. Київ : ДП «УкрНДНЦ», 2023.

4. ISO/IEC 29100:2011. Information technology. Security techniques. Privacy framework. Geneva : International Organization for Standardization, 2011.

5. ISO/IEC 24760-1:2019. IT Security and Privacy. A framework for identity management. Part 1: Terminology and concepts. Geneva : ISO, 2019.

6. ISO/IEC 14443-1:2018. Cards and security devices for personal identification. Contactless proximity objects. Part 1: Physical characteristics. Geneva : ISO, 2018.

7. ISO/IEC 18092:2013. Information technology. Telecommunications and information exchange between systems. Near Field Communication. Interface and Protocol. Geneva : ISO, 2013.

8. ISO/IEC 7816-4:2020. Identification cards. Integrated circuit cards. Part 4: Organization, security and commands for interchange. Geneva : ISO, 2020.

9. NIST. Digital Identity Guidelines: Authentication and Lifecycle Management. Special Publication 800-63B. Gaithersburg : National Institute of Standards and Technology, 2017.

10. OWASP Foundation. Authentication Cheat Sheet. URL: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html.

11. OWASP Foundation. Password Storage Cheat Sheet. URL: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html.

12. OWASP Foundation. Transport Layer Security Cheat Sheet. URL: https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Security_Cheat_Sheet.html.

13. HID Global. HID Mobile Access. Mobile access control solutions. URL: https://www.hidglobal.com/solutions/mobile-access-solutions.

14. HID Global. Seos credential technology. URL: https://www.hidglobal.com/solutions/seos.

15. SALTO Systems. SALTO KS Digital Key. URL: https://support.saltosystems.com/ks/mobile/saltoks-mobile-app/access/.

16. SALTO Systems. Enabling Digital Key. URL: https://saltoks.com/support/enabling-digital-key/.

17. Kisi. Mobile credentials. Product Documentation. URL: https://docs.kisi.io/dashboard/credentials/mobile_credentials/.

18. Kisi. Digital credentials. Product Documentation. URL: https://docs.kisi.io/dashboard/credentials/digital_credentials/.

19. Brivo. Mobile Access Control. URL: https://www.brivo.com/mobile-access-control.

20. Avigilon Alta / Openpath. How to create a mobile credential. URL: https://openpath.atlassian.net/wiki/spaces/EHC/pages/760185611/How+do+I+create+a+Mobile+Credential.

21. Avigilon. Access Control System Integration Guide. URL: https://www.avigilon.com/.

22. SQLite Consortium. SQLite Documentation. URL: https://www.sqlite.org/docs.html.

23. PostgreSQL Global Development Group. PostgreSQL Documentation. URL: https://www.postgresql.org/docs/.

24. Node.js Foundation. Node.js Documentation. URL: https://nodejs.org/en/docs.

25. Python Software Foundation. Python 3 Documentation. URL: https://docs.python.org/3/.

26. Sommerville I. Software Engineering. 10th ed. Boston : Pearson, 2016. 816 p.

27. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner’s Approach. 9th ed. New York : McGraw-Hill Education, 2019. 704 p.

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

ДОДАТОК А

Фрагмент SQL-коду створення фізичної моделі бази даних

```
CREATE TABLE roles (
    role_id INTEGER PRIMARY KEY AUTOINCREMENT,
    role_name VARCHAR(50) NOT NULL UNIQUE,
    role_description VARCHAR(150)
);

CREATE TABLE users (
    user_id INTEGER PRIMARY KEY AUTOINCREMENT,
    role_id INTEGER NOT NULL,
    full_name VARCHAR(100) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    phone VARCHAR(20),
    user_status VARCHAR(20) NOT NULL,
    FOREIGN KEY (role_id) REFERENCES roles(role_id)
);

CREATE TABLE digital_keys (
    key_id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL,
    key_code VARCHAR(80) NOT NULL UNIQUE,
    activation_date DATETIME NOT NULL,
    expiry_date DATETIME NOT NULL,
    auth_method VARCHAR(30) NOT NULL,
    key_status VARCHAR(20) NOT NULL,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

CREATE TABLE buildings (
    building_id INTEGER PRIMARY KEY AUTOINCREMENT,
    building_name VARCHAR(100) NOT NULL,
    address VARCHAR(150) NOT NULL,
    building_type VARCHAR(50) NOT NULL
);

CREATE TABLE access_zones (
    zone_id INTEGER PRIMARY KEY AUTOINCREMENT,
    building_id INTEGER NOT NULL,
    zone_name VARCHAR(100) NOT NULL,
    security_level INTEGER NOT NULL,
    zone_status VARCHAR(20) NOT NULL,
    FOREIGN KEY (building_id) REFERENCES buildings(building_id)
);

CREATE TABLE access_devices (
    device_id INTEGER PRIMARY KEY AUTOINCREMENT,
    zone_id INTEGER NOT NULL,
    device_name VARCHAR(100) NOT NULL,
    device_type VARCHAR(50) NOT NULL,
    installation_point VARCHAR(100),
    device_status VARCHAR(20) NOT NULL,
    FOREIGN KEY (zone_id) REFERENCES access_zones(zone_id)
);

CREATE TABLE access_permissions (
    permission_id INTEGER PRIMARY KEY AUTOINCREMENT,
    key_id INTEGER NOT NULL,
```

```
    zone_id INTEGER NOT NULL,
    granted_by INTEGER NOT NULL,
    granted_date DATETIME NOT NULL,
    end_date DATETIME,
    permission_status VARCHAR(20) NOT NULL,
    FOREIGN KEY (key_id) REFERENCES digital_keys(key_id),
    FOREIGN KEY (zone_id) REFERENCES access_zones(zone_id),
    FOREIGN KEY (granted_by) REFERENCES users(user_id)
);

CREATE TABLE access_events (
    event_id INTEGER PRIMARY KEY AUTOINCREMENT,
    key_id INTEGER NOT NULL,
    device_id INTEGER NOT NULL,
    event_time DATETIME NOT NULL,
    access_result VARCHAR(20) NOT NULL,
    denial_reason VARCHAR(150),
    direction VARCHAR(20) NOT NULL,
    FOREIGN KEY (key_id) REFERENCES digital_keys(key_id),
    FOREIGN KEY (device_id) REFERENCES access_devices(device_id)
);
```

ДОДАТОК Б

Фрагмент програмного коду перевірки цифрового ключа

```
from datetime import datetime

class AccessService:
    def __init__(self, repository):
        self.repository = repository


     def check_access(self, key_code: str, zone_id: int, device_id: int) ->
dict:
        key = self.repository.find_key_by_code(key_code)
        if key is None:
            return self._deny(None, device_id, 'ключ не знайдено')

        if key['key_status'] != 'active':
            return self._deny(key['key_id'], device_id, 'ключ неактивний')

        now = datetime.now()
        if now < key['activation_date'] or now > key['expiry_date']:
               return self._deny(key['key_id'], device_id, 'строк дії ключа
завершено')

        permission = self.repository.find_active_permission(
            key_id=key['key_id'],
            zone_id=zone_id,
            current_time=now
        )

        if permission is None:
              return self._deny(key['key_id'], device_id, 'немає дозволу для
зони')

        self.repository.create_access_event(
            key_id=key['key_id'],
            device_id=device_id,
            result='granted',
            reason=None,
            direction='entry'
        )

        return {
            'allowed': True,
            'message': 'доступ надано',
            'key_id': key['key_id'],
            'zone_id': zone_id
        }

     def _deny(self, key_id: int | None, device_id: int, reason: str) ->
dict:
        self.repository.create_access_event(
            key_id=key_id,
            device_id=device_id,
            result='denied',
            reason=reason,
            direction='entry'
        )
        return {
```

```
    'allowed': False,
    'message': reason
}
```

ДОДАТОК В

Фрагмент програмного коду журналювання подій доступу

```
class EventJournalService:
    def __init__(self, repository, notification_service):
        self.repository = repository
        self.notification_service = notification_service

    def register_event(self, event: dict) -> None:
        self.repository.save_event(event)

        if self._is_critical(event):
            message = self._build_notification(event)
            self.notification_service.send_to_administrator(message)

    def _is_critical(self, event: dict) -> bool:
        if event['access_result'] == 'denied':
            attempts = self.repository.count_recent_denials(
                device_id=event['device_id'],
                minutes=10
            )
            return attempts >= 3

        if event.get('device_status') == 'offline':
            return True

        return False

    def _build_notification(self, event: dict) -> str:
        return (
            f"Критична подія доступу: пристрій {event['device_id']}, "
            f"результат {event['access_result']}, "
            f"час {event['event_time']}"
        )
```

ДОДАТОК Г

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

ДЕКЛАРАЦІЯ ЗДОБУВАЧА

Я, Сердюк Владислав, здобувач(ка) НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота (проєкт) на тему «Програмне забезпечення інформаційної системи агрохімічного паспорта полів» 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 (обрати необхідне):
    - [ ] інструменти генеративного ШІ не використовувалися.
    - [+] інструменти ШІ (вказати назву: ChatGPT, Gemini, Claude, DeepL, _____інші (вказати назву)) були використані для:
        - [ ] перекладу іншомовних джерел;
        - [ ] 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. _____ (підпис) **Владислав Сердюк**