Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інформаційних технологій та комп'ютерної інженерії
(повна назва інституту, назва факультету (відділення))
Кафедра обчислювальної техніки
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до комплексної магістерської кваліфікаційної роботи
магістр
(освітньо-кваліфікаційний рівень)

на тему: "Комплекс мобільного керування автоматикою пропуску. Частина 2. "Програмна частина"

08-54.КМКР.008.00.000 ПЗ

Виконала: студентка групи 2КІ-22м
спеціальності
123 – Комп'ютерна інженерія
(шифр та назва напряму підготовки, спеціальності)
Зубринська Д.Л.,
(прізвище та ініціали)
Керівник: к.т.н., доц. каф.ОТ
Городецька О.С.,
(прізвище та ініціали)
Опонент: к.т.н., доц.каф.МБІС
Карпінець В.В.
(прізвище та ініціали)

Допущено до захисту

Завідувач кафедри ОТ

д.т.н., проф. Азаров О.Д.

12 06 2024 року

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

Освітньо-кваліфікаційний рівень —магістр

Спеціальність 123 — «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д. т. н , проф. Азаров О. Д.

«12» 03 2024 р.

З А В Д А Н Н Я

НА КОМПЛЕКСНУ МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студентці_____ Зубринській Діані Леонідівні_____

(прізвище, ім'я, по батькові)

1 Тема роботи: "Комплекс мобільного керування автоматикою пропуску". Частина 2. Програмна частина", керівник роботи: к.т.н., доцент кафедри ОТ Городецька О. С., затверджена наказом Вінницького національного технічного університету від 11.03.2024 року № 81.

2 Строк подання студентом роботи: 24.05.2024

3 Вихідні дані до роботи: системи контролю і управління доступом, комплекс дистанційного керування автоматикою пропуску з використанням мобільного застосунку на смартфоні, застосунок адміністратора, веб-сервер, тестування й комплексна перевірка функціонування програмного забезпечення.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): огляд і аналіз програмно-апаратних засобів керування доступом; вибір технологій проектування програмного забезпечення; розробка бази даних; дослідження й тестування розробленого програмного забезпечення; розрахунок економічних показників.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) : алгоритми роботи, UML-діаграми, лістинг модулей та ілюстративний матеріал щодо розробки програмної частини.

6 Консультанти розділів роботи наведені в таблиці 1.

Таблиця 1 – Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1,2,3,4	Крупельницький Л. В., к.т.н., доцент каф. ОТ		
5	Небава М.І., к.е.н., професор каф. ЕПВМ		

7 Дата видачі завдання 01.03.2024

8 Календарний план виконання МКР наведено в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів дипломної магістерської роботи	Строк виконання етапів роботи	Примітка
1	Огляд і аналіз програмно-апаратних засобів керування допуском	30.03.2024	Виконано
2	Вибір технологій розробки програмної частини	20.04.2024	Виконано
3	Розробка клієнтського й серверного програмного забезпечення	10.05.2024	Виконано
4	Дослідження й тестування розробленого програмного забезпечення	20.05.24	Виконано
5	Економічна частина	24.05.2024	Виконано

Студентка _____ Зубринська Д.Л.
(підпис) (прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи _____ Городецька О.С.
(підпис) (прізвище та ініціали)

Консультант з економічної частини _____ Небава М.І.

АНОТАЦІЯ

УДК 004.4

Зубринська Д.Л. "Комплекс мобільного керування автоматикою пропуску". Частина 2. "Програмна частина" Комплексна магістерська кваліфікаційна робота зі спеціальності 123 - Комп'ютерна Інженерія, Вінниця: ВНТУ, 2024.

На укр. мові. Бібліогр.: 108 сторінок, 5 розділів; 20 рисунків, 15 таблиць, 18 джерел за переліком посилань.

У роботі розглянуто принципи побудови системи віддаленого управління доступом за мобільним ідентифікатором, що надається віддалено через Інтернет. В роботі проведений аналіз сучасних технологій управління доступом, розглянуті основні принципи побудови систем управління доступом, розглянуто принципи та технології віддаленого керування доступом, визначено підходи до розробки програмного забезпечення бази даних системи віддаленого керування доступом з розширеними функціональними можливостями і підвищеною захищенністю. Розроблено програмне забезпечення бази даних веб-сервера. Проведено тестування й комплексну перевірку розробленого комплекса.

Ключові слова: управління доступом, мобільна ідентифікація, база даних, хмарний сервер, веб-додаток .

ABSTRACT

Zubrynska Diana, L. "Mobile Control System for Access Automation". Part 2. "Software Component" Comprehensive Master's Qualification Thesis in the specialty 123 - Computer Engineering, Vinnytsia: VNTU, 2024.

In Ukrainian. Bibliography: 108 pages, 5 sections; 20 figures, 15 tables, 18 sources listed in the references.

The work examines the principles of building a remote access control system using a mobile identifier provided remotely over the Internet. An analysis of modern access control technologies is conducted, covering the basic principles of access control system construction. The principles and technologies of remote access management are discussed, along with approaches to developing database software for a remote access control system with enhanced functionality and increased security. Software for the database of the web server is developed, followed by testing and comprehensive verification of the developed system.

Keywords: access control, mobile identification, database, cloud server, web application

ЗМІСТ

ВСТУП

Сучасні системи управління доступом вирішують широкий спектр завдань, включаючи обмеження проникнення людей і транспорту на територію об'єкта, що охороняється, ведення обліку робочого часу персоналу та контроль його переміщення у певний час доби, а також обмеження доступу до окремих зон [1].

Актуальність дослідження випливає з необхідності забезпечення безпечного функціонування різноманітних об'єктів, таких як бізнес-підприємства, житлові будівлі, виробництво та соціальні установи. Один із найбільш ефективних та цивілізованих підходів до забезпечення комплексної безпеки різних типів об'єктів полягає у використанні систем контролю та управління доступом. Такі системи дозволяють ефективно управляти несанкціонованим доступом до території, будівель, окремих поверхів та приміщень, не перешкоджаючи при цьому проходженню персоналу та відвідувачів у відведені для них зони.

Однією з основних вимог до сучасних систем управління доступом є досягнення максимальної ефективності та зниження впливу людського фактору. Внаслідок цього системи управління доступом стали одними з найбільш розвинених сегментів ринку безпеки. Навіть під час кризових періодів падіння в цьому сегменті ринку було незначним. Це пояснюється тим, що системи управління доступом не лише сприяють підвищенню рівня запобігання потенційним загрозам, а й приносять значний економічний ефект [2].

Постійний розвиток індустрії безпеки сьогодні визначається не лише розширенням глобальної економіки, але й активізацією окремих галузей, таких як роздрібна торгівля, готельний бізнес, транспортна сфера та будівництво, які використовують такі системи. Системи управління доступом, що є важливою складовою безпеки, мають великий потенціал, оскільки можуть взаємодіяти та інтегруватися з іншими інформаційними системами, зокрема з системами відеоспостереження, охоронною сигналізацією та системами управління персоналом [3]

Об’єктом дослідження є процеси віддаленого керування доступом з використанням технологій Інтернет.

Предметом дослідження є серверні та клієнтські програмні засоби бази даних мобільних користувачів для керування пропуском в системі доступу.

Метою роботи є розширення функціональних можливостей комплексу мобільного керування автоматикою пропуску з використанням безпровідних технологій, інтегрованих з програмним забезпеченням смартфонів та інтернет-мережі.

Для досягнення мети в роботі потрібно вирішити такі основні завдання:

— огляд і аналіз програмно-апаратних засобів керування доступом;

— аналіз та вибір технологій проектування програмного забезпечення комплексу керування пропуском;

— розробка бази даних керування пропуском для серверного програмного забезпечення

— адаптація застосунку користувача до розробленої бази даних;

— комплексне дослідження й тестування розробленого програмного продукту;

— розрахунок економічних показників проєкту.

Для досягнення мети роботи використовуються такі **методи дослідження:**

— системний аналіз;

— методи алгоритмічного проектування;

— методи програмування мобільних застосунків.

Новизна дослідження полягає в тому, що набула подальшого розвитку технологія віддаленого мобільного програмного керування автоматикою пропуску на об’єкти, яка відрізняється від аналогів збільшеним захистом та розширеними функціональними можливостями за рахунок створення спеціалізованої бази даних.

Практичне значення розробленого програмного забезпечення полягає в тому, що його збільшена захищенність і розширена функціональність дозволяє створювати сучасні системи керування пропуском на різні об’єкти особистого, громадського та спеціалізованого призначення.

Апробація в доповіді та публікація тез за темою роботи:

АПАРАТНО-ПРОГРАМНИЙ КОМПЛЕКС МОБІЛЬНОГО ДИСТАНЦІЙНОГО КЕРУВАННЯ ДОСТУПОМ / Гуменюк В.В., Зубринська Д. Л., Крупельницький Л.В., Городецька О.С // Міжнародна науково-практична Інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (МН2024)» (15.10.2023 р. – 20.05.2024 р., Вінниця) : Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/21172/17557

1 ОГЛЯД І АНАЛІЗ ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ КЕРУВАННЯ ДОСТУПОМ

1.1 Оснвні функції СКУД

СКУД — системи контролю та управління доступом (англ. — Physical Access Control System — PACS). Загалом, СКУД —- це електронна система, що фізично обмежує доступ будь-куди, наприклад, на вхід і вихід з території чи приміщення, що охороняється.

Сучасна СКУД — автоматизована система пропуску, яка використовується для управління доступом людей або транспортних засобів на територію об'єкта . Це може бути ціла офісна будівля, виробничий цех, склад, будівельний майданчик, паркувальний майданчик або звичайна квартира. Люди, які отримують доступ, можуть бути власниками, співробітниками, підрядниками, обслуговуючим персоналом або відвідувачами.

Роботи будь-якого СКУД можна представляти у вигляді 5 логічних етапів.

Авторизація — процес надання дозволу на вхід у будівлю або певні приміщення на території у певний час. Адміністратор СКУД видає чи не видає дозволу на доступ, ґрунтуючись на різних критеріях, у тому числі на тому, чи є людина співробітником, підрядниками чи відвідувачами, а також ґрунтуючись на формальних процедурах перевірки особи. Наприклад, сканери Regula, що інтегровані з програмним забезпеченням СКУД Parsec [1], не лише автоматизують процес введення паспортних даних у СКУД, а й автоматично перевіряють паспорт на справжність. Результатом авторизації є видача ідентифікатора.

Аутентифікація, при вході до приміщення відвідувач прикладає до зчитувача ідентифікатор, який він отримав на етапі авторизації. Ідентифікатор може представляти собою RFID-карту, пін-код, віртуальний ідентифікатор на смартфоні, QR-код або відбиток пальця. Ідентифікатор дозволяє СКУД розпізнати його і підтвердити, чи має цей ідентифікатор право доступу чи ні.

Далі потрібно підтвердити, що пред'явник ідентифікатора - це та ж сама особа, яка проходила авторизацію. Підтвердити особу власника найпростіше за

допомогою системи розпізнавання обличчя, що відбувається шляхом порівняння зображення обличчя пред'явника ідентифікатора, захопленого відеокамерою, із зображенням, що зберігається в базі даних програмного забезпечення СКУД. Наприклад, така функція присутня в СКУД Sigur у вигляді модуля програмного забезпечення Sigur верифікація обличчя. Таке підтвердження не потрібне, якщо використовується біометрична СКУД. Біометричну ідентифікацію часто називають чистою або реальною аутентифікацією, оскільки використовується дійсно невідчужувана біометрична ознака (ідентифікатор), що має пряме відношення до людини

Доступ, якщо ідентифікатор підтверджено і людина має відповідні дозволи на доступ, то на двері, ворота, турнікет, ліфт або іншу точку входу посилається електронний вихідний сигнал, який розблоковує двері і дозволяє йому увійти. Також на цьому етапі система логує дані про те, хто куди і коли увійшов і вийшов, включаючи дані про тих, хто намагався отримати доступ до приміщення, не маючи на це прав доступу.

Управління і моніторинг — адміністратори СКУД можуть постійно додавати, видаляти або змінювати дозволи залежно від змін потреб компанії, а також від того, хто і в який час повинен перебувати на території. Крім того, адміністратори стежать за електронними журналами входу/виходу, щоб переконатися, що доступ на територію мають лише авторизовані користувачі, і бути в курсі всіх загроз безпеці.

Зрозуміло, що спостереження за всіма подіями системи недоцільне, оскільки це вимагатиме занадто великої кількості ресурсів. При проектуванні системи повинні бути передбачені можливості для детекції подій, які компанія вважає для себе тривожними, та сповіщення відповідальних осіб про такі події. Для цього в сучасних СКУД передбачені програмні модулі, які дозволяють гнучко налаштовувати реакції на події в системі СКУД та оперативно отримувати сповіщення про них у Telegram, Viber, SMS або на електронну пошту.

Звітність , при виникненні загрози безпеці (або навіть підозрілої активності) адміністраторам СКУД та співробітникам служби безпеки необхідно уважно вивчити журнали доступу. І за потреби передавати дані владі. Термін зберігання журналів доступу та інших пов'язаних з ними даних повинен визначатися компаніями, виходячи з їхніх політик у сфері безпеки та вимог нормативних документів, яким вони повинні відповідати.
Крім цього, звітність використовується в інших підрозділах компанії, наприклад звіти від відпрацьованого часу можуть передаватися керівництву для оцінки відвідуваності, або в бухгалтерію для нарахування зарплати відповідно до відпрацьованого часу.

1.2 Переваги застосування систем керування доступом

Безпека підприємства – найважливіший фактор про те, що СКУД є стримуючим фактором для злочинців, ідея тут проста — чим більше зусиль потрібно зробити порушнику, тим більша ймовірність відмови від скоєння злочину. Є цікаве дослідження, яке дозволяє приблизно оцінити розмір ефекту. У Великій Британії було встановлено, що встановлення воріт і хвірток при в'їзді в провулок і вході в провулок знизило кількість крадіжок зі зломом на 43%. Часто доводиться чути, що типу, «у нас не банк, нам побоюватися нічого», так ось колеги зі США наводять приголомшливі цифри 30% банкрутств у бізнесі викликані крадіжкою співробітників , а 68% крадіжки випадків відбуваються в малому та середньому бізнесі .

Система контролю доступу відіграє важливу роль в інтегрованій системі безпеки об'єкта, забезпечуючи захист співробітників, будівель і майна. Вона може працювати в комплексі з іншими пристроями безпеки, присутніми на об'єкті, і можливість інтеграції детально розглянута нижче.

Бізнес-аналітика, у поєднанні з функціями обліку робочого часу система контролю доступу стає справжньою аналітичною системою для вашої організації, і облік робочого часу - не єдиний інструмент для цього. Широкі

можливості інтеграції з системами документообігу, HR-системами, бухгалтерськими системами, CRM-системами тощо виводять автоматизований збір та обробку даних на новий рівень.

Систему контролю доступу можна інтегрувати практично в будь-яку систему, яку підприємство використовує для основної діяльності CRM, ERP, системи кадрового обліку або інші. Організувати взаємодію із системою управління підприємством можна на програмному рівні з використанням API або SDK сервісів (їх безкоштовно надає більшість виробників СКУД).

Скорочення постійних витрат — СКУД непросто система доступу, це автоматизована система, тобто. система, яка повинна мінімізувати участь людей у роботі. Тому що люди це завжди помилки і завжди зарплата, і те й друге непогано б мінімізувати. Правильно спроектована та встановлена система може значно зменшити витрати на безпеку. Крім того, сучасні системи контролю та управління доступом пропонують великий потенціал для скорочення постійних витрат завдяки автоматизації процесів.

Відома незліченна кількість бойовиків, у яких «погані хлопці» намагалися знищити світ, попередньо захопивши якусь смертельну зброю. І перший крок на цьому сумнівному шляху був у тому, щоб роззброїти охорону яка дрімала на прохідній. Ці часи безповоротно минули, коли просте розміщення охоронця на вході до будівлі було одним із головних способів захисту від небажаного вторгнення. У сучасній парадигмі захисту такий підхід вважається дорогим та неефективним. Сучасні системи реєстрації відвідувачів можуть, наприклад, замінити цілий паспортний стіл, а це щонайменше 2 людини. А за рахунок інтеграції системи обліку робочого часу в системи розрахунку заробітної плати скорочується час, необхідний для нарахування заробітної плати.

Облік робочого часу співробітників, коли обговорюється необхідність обліку робочого часу, зазвичай виникає велика системна помилка. Всі аргументи опонентів зводяться до того, що суворий облік робочого часу погіршує ефективність праці, тобто що, відпрацювавши рівно 8 годин, ви вже нічого не можете робити або можете працювати лише ненормовано, і це дійсно так. Помилковою є думка, що облік робочого часу є лише одним з інструментів,

які впливають на ефективність праці. Правильніше було б сказати, що роль систем обліку робочого часу полягає в тому, щоб показати ефективність вже існуючих систем стимулювання ефективності роботи (системи оплати праці, KPI і т.д.).

Відвідувачі сайту не враховуються, але є важливими для оцінки ефективності сайту та його розвитку. Не забувайте про мотивуючу силу обліку робочого часу та зручність для ваших співробітників.

Не варто розглядати хронометраж як репресивний захід, хоча це зручний спосіб підвищити ефективність роботи з його допомогою.

Комфорт у використанні СКУД — Сучасна СКУД це не тільки і не так, про обмеження, як про керування доступом. Для того щоб організувати комфортний і безпечний рух великої кількості людей на досить обмеженій площаді, необхідні деякі правила руху, інакше буде хаос, який обов'язково призведе до негативних наслідків. Щоб цього не допустити, на вулицях у нас є ПДР, а в будинках є СКУД. Зручним і комфортним має бути не лише пред'явлення ID-картки, а й процедура реєстрації користувачів та видачі ID-карток, зручне носіння ID-карток, обмін ID-карток та інші дрібніші, але не менш важливі процеси. Сучасні програмно-апаратні платформи здатні одночасно підвищити рівень безпеки та комфорту користувачів. Не варто забувати і про організаційні заходи, адже АСУ — це лише інструмент, за допомогою якого можна вбивати сокирою бабусь або будувати будинки.

1.3 Склад типових систем керування доступом

Система контролю та управління доступом зазвичай складається з низки компонентів, починаючи від компонентів ідентифікації користувачів і закінчуючи компонентами прийняття рішень про доступ.

Типовий склад систем:

— пристрої ідентифікації;

— контролери

— програмне забезпечення

— запобіжні пристрої

Пристрої ідентифікації можна розбити на дві логічні групи - ідентифікатори та зчитувачі. Кожна з цих груп має власні відмінні властивості, які необхідно враховувати при виборі СКУД.

Ідентифікатор — це унікальна ознака людини, що дозволяє відрізняти її від інших людей, тобто ідентифікувати. Ну наприклад якщо у вас є RFID брелок, наявність цього брелока у вас і є те, що відрізняє ват від інших об'єктів які ним не володіють, а значить ви маєте право доступу туди куди інші не мають. Ну чи пін-код, це вже ідентифікація на основі знання, ви знаєте пін-код, вводите його та отримуєте доступ. Біометричні характеристики людини також можуть використовуватися як ідентифікатори. Найпоширенішими біометричними системами контролю доступу є відбитки пальців, візерунок вен на пальцях або руці, геометрія обличчя та райдужної оболонки ока.

Найпоширеніші ідентифікатори в сучасних СКУД базуються на радіочастотній технології безконтактної передачі даних RFID. Наразі 90% систем контролю доступу працюють саме з ними. Найпоширенішими формами факторів є пластикова картка і брелок, хоча, звичайно, існують також браслети і цілий ряд міток. Головні плюси — низька ціна, механічна міцність, ні чим не обмежений термін життя, відсутність активного джерела живлення.

Найпоширеніші безконтактні карти RFID ідентифікатори в Україні — це HID Prox і EM-marine , Mifare Classic та Mifare Ultralight . Усіх їх поєднує те, що вони слабо захищені від копіювання, слабо захищені від технічного злому .

Справжній захист від копіювання та підробки забезпечують ідентифікатори, у чіпах яких реалізовано криптографічний захист. Це безконтактні смарт-карти, що працюють на частоті 13,56 МГц. Найбільш поширеними в Україні є карти Mifare Plus , Mifare DESFire .

Ідентифікація по смартфону — віртуальні ідентифікатори використовуються разом зі смартфоном і додатком для доступу, який має бути встановлений на смартфоні. Це випадково згенероване числове значення, яке видається при реєстрації користувача. Залежно від провайдера, можуть бути як безкоштовні, так і платні ідентифікатори.

Віртуальні ідентифікатори — ідентифікація за номером мобільного телефон. Можна використовувати номер мобільного телефону як ідентифікатор. Оскільки всі номери унікальні, для їх використання потрібен GSM-контролер або GSM-зчитувач.

GSM-зчитувачі використовуються для зчитування телефонних номерів.

З точки зору безпеки, номер мобільного телефону є поганим варіантом, оскільки існує багато способів підробити номер і ще більше способів дізнатися ваш номер мобільного телефону.

Ідентифікація за допомогою пін-коду. ПІН-код можна використовувати як повноцінний ідентифікатор. Для його введення необхідно використовувати зчитувач з кодовою клавіатурою. Як правило, такий зчитувач поєднаний з RFID-зчитувачем в одному корпусі. ПІН-код - це єдиний ідентифікаційний елемент, який не може бути використаний без вашої добровільної згоди на його пред'явлення. Іншими словами, картку можна вкрасти, і навіть будь-який біометричний ідентифікатор можна нанести на зчитувач проти волі власника, наприклад, просто піднісши палець до зчитувача відбитків пальців. З PIN-кодом, однак, такі методи неефективні.

Зчитувачі з кодовим полем для введення пін-коду – це один з небагатьох ідентифікаторів, який неможливо вкрасти, залишити вдома або загубити. Однак його можна забути або легко передати іншій особі. Ви можете використовувати

PIN-код як окремий ідентифікатор або в режимі PIN + RFID-картка. У цьому випадку доступ надається після послідовного пред'явлення обох ідентифікаторів.

Біометрична ідентифікація — біометричні ідентифікатори мають багато переваг: вони безкоштовні, їх неможливо забути, загубити або зламати. Як правило, їх важко підробити. В даний час в СКУД використовуються наступні типи біометричних ідентифікаторів: Відбиток пальця, райдужна оболонка ока, геометрія обличчя, малюнок вен. З усіх можливих біометричних ідентифікаторів, відбиток пальця є одним з найпоширеніших біометричних об'єктів для використання в СКУД.

Описані вище біометричні ідентифікатори потребують біометричних зчитувачів. Біометричні технології засновані на «зчитуванні» певних фізичних характеристик користувача. Дані перетворюються на унікальний код, який потім надсилається до бази даних системи контролю доступу та програмного забезпечення.

Біометрична ознака, наприклад, відбиток пальця, зберігається в базі даних користувача у вигляді цифрової послідовності, з якої неможливо відновити папілярний візерунок пальця. Це дозволяє повністю відповідати закону про захист персональних даних.

Біометричні технології, що використовуються в системах доступу, мають ряд загальних переваг перед традиційними системами, заснованими на використанні токенів та/або персональних ідентифікаційних кодів - Підвищена безпека в порівнянні з традиційними системами ідентифікації; відсутність можливості передачі біометричних даних іншій особі, як це часто буває в системах доступу на основі токенів і кодів; Зниження ризику шахрайства на пропускних пунктах і робочих місцях (при використанні в системах обліку робочого часу); зниження ризиків, пов'язаних з втратою токенів і втратою персональних ідентифікаційних кодів; зниження витрат на адміністрування системи за рахунок усунення процесів, пов'язаних з роботою з втраченими або вкраденими бейджами;

Більшість зчитувачів здатні надавати звуковий та візуальний зворотній зв'язок, щоб користувач знав, чи отримав він доступ чи ні. Існують різні типи

зчитувачів в СКУД, які використовують різні методи запису, зберігання та зчитування кодової інформації, забезпечують різні рівні безпеки та мають значну ціну: зчитувачі банківських карток та зчитувачі контактних ключів з сенсорною пам'яттю

Пристрої безпеки являють собою фізичний бар'єр, який перешкоджає доступу до контрольованої зони. Всі пристрої можна розділити на дві групи залежно від принципу дії: електромеханічні та електромагнітні.

Унікальною особливістю турнікетів, якої позбавлені всі інші загороджувальні пристрої, є можливість відсікати пасажирів індивідуально. Це робить їх практично незамінними, коли мова йде про ідентифікацію кожної людини, що входить в будівлю.

Електричні замки, найважливіше, що потрібно знати про електричні замки — це те, що вони мають різні рівні безпеки, від електричних засувів найнижчого класу безпеки до електричних замків з електроприводом найвищого класу безпеки.

Контролери — це інтелектуальні мікропроцесорні пристрої. Контролер виконує наступні завдання: Зберігання бази даних користувачів, ведення журналу подій, управління системами шлагбаумів і багато іншого. Саме контролер приймає рішення про надання доступу конкретному користувачеві через конкретні двері в конкретний час доби.

Контролер СКУД здатний працювати в автономному режимі, якщо з якихось причин відсутнє з'єднання з мережею передачі даних. Мережеві контролери підключаються до персональних комп'ютерів, на яких встановлено програмне забезпечення, зазвичай розроблене тією ж компанією, що і контролер. Без перебільшення можна сказати, що контролер - це мозок будь-якої системи доступу.

Контролери бувають автономні та мережеві. Автономні контролери, як випливає з назви, працюють виключно в автономному режимі, тобто їх не можна підключити до інших контролерів. Якщо на вашому об'єкті встановлено кілька автономних контролерів, вам потрібно буде відвідувати кожен окремий

контролер, щоб перевести його в режим програмування і додати або видалити карти. Основною перевагою автономних контролерів є їхня низька ціна.

Мережеві контролери, незалежно від їх кількості, можна легко об'єднати в єдину мережу, і вони зазвичай підключаються до комп'ютера з встановленим програмним забезпеченням, з якого можна легко і просто керувати всією системою доступу. Дуже важливий нюанс полягає в тому, що виробник контролера завжди розробляє і програмне забезпечення до нього, і контролери, які він виробив, будуть працювати тільки з програмним забезпеченням, яке він розробив.

Більшість сучасних контролерів передачі даних підключаються безпосередньо до мережі Ethernet. Підключення здійснюється стандартним мережним кабелем кручена пара, для підключення якого плата контролера забезпечена стандартним Ethernet-роз'ємом RJ-45. До переваг такого підключення можна віднести те, що ви можете використовувати вже існуючу в компанії мережу, можна підводити живлення не окремим кабелем по тій же кручений парі з стандарту Ethernet з використанням технології PoE (Power over Ethernet), до недоліків те, що максимальна відстань від контролера до точки підключення до мережі 100 метрів. Виділяються мережеві контролери з підключенням по Ethernet і по RS-485.

Системне програмне забезпечення для СКУД може бути різним за розміром — від простого автономного рішення, встановленого на одному комп'ютері в незахищеному приміщенні, до потужного мережевого пакета, встановленого на сервері в приміщенні з контрольованим доступом.

Кожне програмне забезпечення можна класифікувати за кількома критеріями.

Ціна програмного забезпечення Платне програмне забезпечення — це класичний і найпопулярніший підхід, оскільки, як ми знаємо з прикладу Microsoft, продаж програмного забезпечення приносить великі прибутки.

Окремо слід згадати про програмне забезпечення, що постачається за допомогою хмарних технологій, оскільки воно обіцяє великі переваги. Цей тип програмного забезпечення надзвичайно простий і зручний у використанні,

оскільки для його використання не потрібно нічого встановлювати, ви просто даєте користувачеві IP-адресу, встановлюєте логін і пароль для доступу. І вся робоча станція готова до використання.

Для роботи з програмним забезпеченням можна використовувати будь-який браузер. Ще однією перевагою є кросплатформенна сумісність клієнтських комп'ютерів, оскільки браузер чудово працює на операційних системах, заснованих на ядрі Linux, яке, як правило, є безкоштовним.

1.4 Особливості й переваги комплексу, що розробляється над аналогами

Розглянемо позиції відомих зарубіжних виробників аналогічних систем СКУД до комплексу керування пропуском, що розробляється.

Компанія HID Global одна з найбільших компаній світових компаній у галузі безпеки. HID Global входить до складу групи компаній ASSA ABLOY Group, річний оборот якої складає близько 1,1 млрд. доларів.

Компанія NedAp має річний оборот близько 200 млн. доларів Голландська компанія є одним з провідних виробників систем радіочастотної ідентифікації. Компанія була заснована в 1929 році і спеціалізується на виробництві електронних систем автоматичної ідентифікації. NedAp - публічна компанія, її акції торгуються на біржі.

Компанія Suprema (оборот біля 50 млн. доларів) Корейська компанія, що спеціалізується на біометричних системах доступу. Suprema - публічна компанія, її акції торгуються на корейському біржовому торговому майданчику.

Компанія ZKTeco має більше 2000 показів на місяць Китайська компанія ZKTeco була заснована в 1985 році і спеціалізувалася на біометричних системах доступу, наразі компанія виробляє повний спектр обладнання для систем безпеки. Але біометричні системи досі залишаються флагманом продажів, завдяки низькій ціні та прийнятному рівню якості. Виробничі площі становлять понад 50 000 кв. метрів.

Розроблювана мобільна система контролю доступу LOKKYY (далі система LOKKYY) призначена для використання з різними системами автоматичного контролю доступу транспортних засобів і людей: Ворота, шлагбауми, хвіртки, ролети тощо.

Суттєвою відмінністю комплексу LOKKYY від інших пристроїв дистанційного керування є його універсальність та мобільність завдяки можливості контролю та управління через мережу Інтернет, Bluetooth, Wi-Fi за допомогою відповідного мобільного додатку LOKKYY APP, що встановлюється на смартфони користувачів. Це забезпечує відсутність апаратних ключів, відсутність додаткового апаратного сервера (пристрою централізованого управління), легке забезпечення довгострокового і короткострокового доступу, швидку заміну ключів і розширені функції дистанційного керування і моніторингу.

Комплекс LOKKYY складається з апаратного мікроконтролерного пристрою LOKKYY з вбудованими бездротовими інтерфейсами Wi-Fi, Bluetooth (далі пристрій LOKKYY) та мобільного програмного додатку LOKKYY APP, який завантажується на смартфони користувачів (далі - додаток LOKKYY APP).

Структурна схема розроблюваного комплексу LOKKYY, яка пояснює підключення та взаємодію компонентів для керованого виконавчого механізму, наведена на рис. 1.1.

Пристрій LOKKYY використовує сигнал від магнітного датчика для визначення поточного стану контрольованого об'єкта - відкритий чи закритий - і може керувати автоматикою приводу за допомогою вбудованого реле. Користувач отримує інформацію від пристрою LOKKYY через свій смартфон за допомогою додатку LOKKYY APP з Інтернету та хмарного сервера і може

відстежувати поточний стан та керувати проходом.

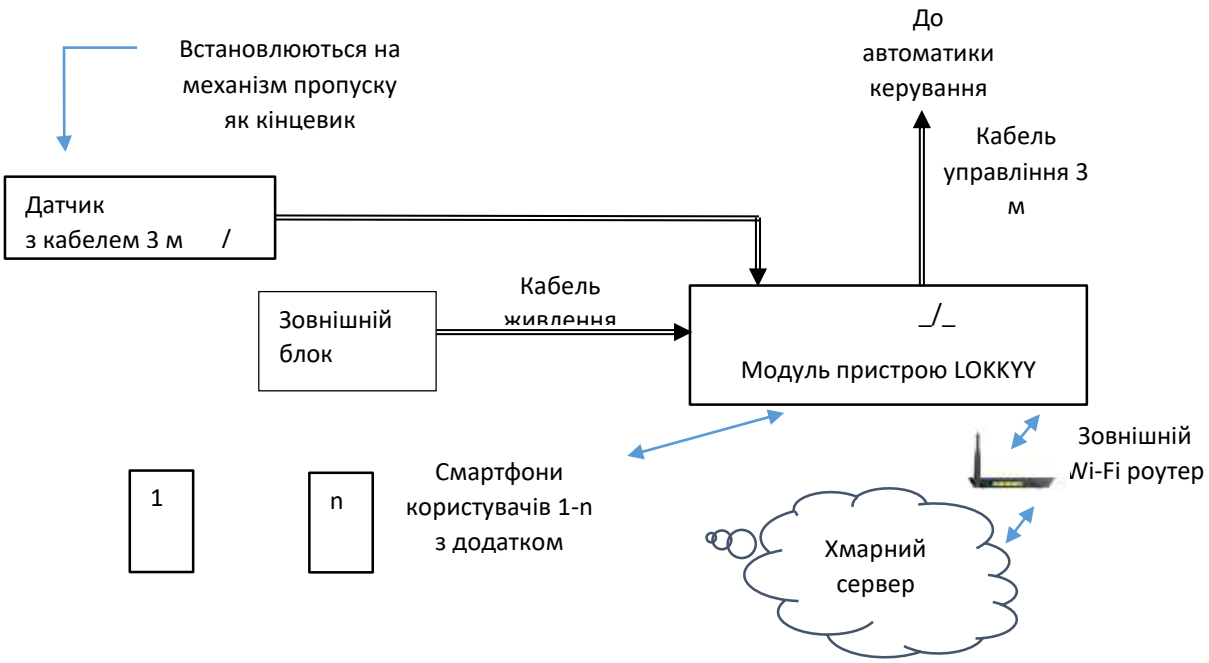


Рисунок 1.1 — Загальна структурна схема комплексу LOKKYY

Пристрій LOKKYY налаштовується зі смартфона адміністратора через бездротову мережу Bluetooth. Пристрій LOKKYY отримує доступ до Інтернету через бездротову мережу Wi-Fi, доступну на контрольному пункті. Якщо Інтернет недоступний, смартфони та пристрій LOKKYY зв'язуються через інтерфейс Bluetooth.

2 ТЕХНОЛОГІЇ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМПЛЕКСУ КЕРУВАННЯ АВТОМАТИКОЮ ПРОПУСКУ

2.1 Аналіз вимог до реалізації програмного комплексу

Розробка присвячена створенню комплексної систему, яка дозволить використовувати дистанційні керуючі елементи, сумісні зі стандартом ISO/IEC 14443a широко відомі як RFID-картки (RFID-картки здатні виконувати складні дії, такі як криптографічна перевірка особистості або локальна зберігання даних. Наразі підтримується лише зчитування ідентифікатора картки, але комунікаційний стек готовий до розширення) для відмикання дверей та доступу до іншого електронного обладнання (далі точки доступу). Для того, щоб ця система була корисною для університету, система має відповідати вимогам, викладеним нижче.

Надійність, оскільки система може використовуватися для захисту цінних ресурсів, наприклад, комп'ютерних класів або лабораторій, він повинен дозволяти доступ тоді і тільки тоді, коли це необхідно. Ми повинні надати користувачеві причини довіряти цій обіцянці. Точки доступу повинні бути доступними, навіть якщо щось піде не так; зокрема, часткове відключення електроенергії або мережі не повинно перешкоджати системі надавати доступ або спричиняти втрату журналів доступу. Тимчасова несправність сервера не повинна викликати жодних проблем. Крім того, дизайн та реалізація повинні передбачати простий механізм обходу відмови.

Безпека —блокування не повинно допускати нелегітимного доступу. Щоб захистити конфіденційність, журнали або ідентифікатори карток не повинні витікати. Ми не можемо припустити приватний канал зв'язку. Тому вся комунікація в обох напрямках повинна бути автентифікована і зберігати конфіденційність.

Комплекс повинен бути ефективним рішенням для нашого варіанту використання. Воно повинно бути ефективним, навіть якщо варіант використання зміниться в майбутньому.

Розширюваність щоб бути готовими до майбутнього і забезпечити можливість поступового розвитку, все програмне забезпечення і все обладнання повинні бути модульними, з чітко визначеними інтерфейсами і розширюваними.

Функції, які не були реалізовані в першій ітерації, але планується додати в майбутньому: довільна комунікація з карткою, керувати довільними пристроями, а не лише дверними замками, модуль WiFi (для випадків, коли живлення є, а Ethernet — ні).

Простота розробки. У майбутньому систему, швидше за все, розроблятимуть і підтримуватимуть студенти, а не штатні розробники. Тому кодова база повинна бути простою, легкою для розуміння і зміни, інструменти і бібліотеки повинні бути простими у використанні, а накладні витрати на залучення нового розробника до проекту повинні бути мінімальними. Коли це можливо, слід використовувати загальні, добре відомі рішення замість рішень, розроблених власними силами.

Простота використання. Налаштування правил доступу має бути простим і зручним. Це не повинно відбуватися за рахунок загальності. Синхронізація з університетом необхідна електронна інформаційна система, щоб можна було автоматично імпортувати інформацію про картки та групи, такі як "викладачі", "студенти" або "аспіранти".

Система повинна повідомляти оператора, якщо потрібне втручання людини, але прості завдання і передбачувані проблеми повинні вирішуватися автоматично.

Простота розгортання та обслуговування. Розгортання має бути простим і з мінімальними накладними витратами. З апаратного боку, повинна бути можливість використовувати існуючу інфраструктуру, щоб не потребувати додаткових кабелів для зв'язку або живлення. З боку програмного забезпечення

має бути можливим імпорт даних з існуючих джерел (таких як Академічна інформаційна система нашого університету). Заміна будь-яких компонентів, що вийшли з ладу, повинна бути швидкою і не вимагати значної підготовки. Система повинна перевіряти свій стан і автоматично виправляти все, що можна виправити автоматично, наприклад, перезавантажувати пристрій, якщо він заблокований.

Доступність — апаратне забезпечення має бути дешевим у виробництві, а компоненти повинні бути доступними в майбутньому або безболісно замінюваними на новіші альтернативи.

Для того, щоб зробити систему максимально доступною, вона передбачає як апаратні схеми, так і програмне забезпечення у відкритому доступі.

Поведінка при відключенні електроенергії. У разі відключення електроенергії в точках входу/виходу деякі двері повинні залишатися зачиненими (щоб уникнути ризику порушення безпеки), а деякі - відчинятися (наприклад, аварійні виходи). Обидва варіанти можна підтримати, використовуючи різні замки та змінюючи конфігурацію.

Аварійне відкриття — апаратні замки на точках входу/виходу повинні підтримувати ручне відкривання та закривання уповноваженим персоналом. Це корисно в надзвичайних ситуаціях.

Існуючі системи, огляд та порівняння. Рішення, які зараз використовуються в університетах, так як і у більшості комерційних організацій, складається з декількох простих зчитувачів карток і централізованого сервера для прийняття рішень та інтерфейсу управління доступом. Зазвичай вони вимагають спеціальної проводки, запроваджують прив'язку до постачальника через власні протоколи зв'язку і не можуть працювати, коли сервер недоступний. На відміну від них, система що пропонується покладається на стандартну, вже існуючу інфраструктуру, надає специфікацію протоколу зв'язку і бібліотеки для розширення системи, а також гарантує, щ о вона продовжить працювати, коли сервер не буде доступним. Існуючі комерційні рішення є дорогими (зазвичай кілька сотень доларів за одиницю) і через прив'язку до постачальника не можуть

бути замінені на альтернативні. Пропонована система має на меті бути майже на порядок дешевшим і повністю відкритим рішенням.

Ключові принципи функціонування системи наступні. Модульність — поділ функціональності на незалежні модулі з чітко визначеними, простими, мінімальними інтерфейсами спрощує розробку, робить дизайн набагато легшим для сприйняття і мінімізує криву навчання для нового розробника. Справи йдуть краще, якщо ви знаєте, де шукати певну функціональність (і де не варто).

Принцип найменшого здивування — люди є частиною системи. Дизайн повинен відповідати досвіду, очікуванням та ментальним моделям користувача [13]. Якщо дизайн, реалізація або поведінка частини системи є достатньо незрозумілою, щоб здивувати вас, її слід переробити.

Має надаватися пріоритет передбачуваності тих частин системи, які важко спостерігати (наприклад, вбудовані пристрої), щоб вони не приносили неприємних сюрпризів користувачеві.

Стан додає складності системі: якщо щось має стан, це означає, що є код, який ним керує, і розробник повинен відстежувати, як внутрішній стан впливає на систему. Крім того, якщо система має стан, це ускладнює відновлення після збоїв, оскільки стан повинен бути реплікований або іншим чином відтворений. Тому, якщо це можливо, компоненти системи повинні мати мало внутрішнього стану і залежати тільки від явних, чітко визначених, легко реплікованих даних.

2.2 Визначення основних компонентів системи для управління доступом

Система складається з сервера і декількох контролерів. Кожен контролер обслуговує одну точку доступу, зберігає копію правил доступу та виконує їх локально. Сервер надає контролерам оновлення правил і збирає журнали доступу.

Сервер зберігає авторизовану версію правил доступу, збирає журнали, забезпечує оновлення програмного забезпечення та синхронізацію часу для інших пристроїв. Він відстежує стан системи (і повідомляє про це в інтерфейсі

керування). За винятком вмісту бази даних, вона повністю не має статусу. Це спрощує код і робить реплікацію та обхід відмови тривіальними.

Контролер контролює пов'язану з ним точку доступу (наприклад, відмикає двері). Він виконує дії (наприклад, відчиняє двері або реєструє вхід) на основі подій, що спостерігаються (наприклад, пред'явлення картки або відчинення дверей). Він періодично зв'язується з сервером, повідомляючи про його стан і перевіряючи наявність оновлень.

Журнали надсилаються на сервер, а правила та оновлення прошивки можна отримати з сервера. Тому пристрій можна замінити, просто записавши правильний ідентифікатор пристрою та ключ шифрування або на пристрої, або в базі даних.

Зчитувач — видимий користувачеві блок в точках доступу, який зчитує RFID- картки і надає візуальний і звуковий зворотний зв'язок про те, чи надано доступ. До одного контролера можна підключити до двох зчитувачів карток. Надається бібліотека для взаємодії з нашими зчитувачами, тому їх можна використовувати незалежно від контролера.

Апаратне забезпечення — сервер не залежить від апаратного забезпечення - він працює на будь-якому комп'ютері з підтримкою роботи в мережі та середовищем Python. Очікується, що для розгортання буде використовуватися загальне серверне обладнання.

Спроектувано та створено спеціальне обладнання для контролерів та зчитувачів. Вони зосередилися на тому, щоб зробити його доступним, перспективним, розширюваним і дешевим. Схеми та інші документи доступні в репозиторії вихідних кодів.

Щоб спростити встановлення, ми намагалися використовувати наявну інфраструктуру, де це можливо: ми використовуємо Ethernet для зв'язку між сервером і контролером, додавши опціональну технологію Power over Ethernet, тому нам не потрібні додаткові кабелі. За бажанням, ми можемо додати до контролера модуль WiFi для тих випадків, коли електрика є, а зв'язку немає. Ми навіть розробили наші зчитувальні блоки та з'єднувальні кабелі так, щоб їх

можна було легко налаштувати, щоб вони були сумісними з існуючими отворами в стінах.

Правила доступу — рішення про надання доступу залежить від особи користувача, точки доступу, дати, часу та дня тижня. Правила доступу розроблені таким чином, щоб бути простими без шкоди для загальності.

2.3 Технічні вимоги до програмної частини системи

Надійність, відповідно до вимог ТЗ, контролери повинні працювати під час збоїв у роботі мережі . Безперервна робота забезпечується зберіганням та оцінкою правил локально на контролері, а сервер потрібен лише для оновлення локальної копії. Втрати журналів доступу можна уникнути, зберігаючи їх локально, доки вони не будуть збережені на жорсткому диску сервера.

Для підвищення доступності можна розгорнути кілька серверів, якщо резервна база даних синхронізується за допомогою загальних механізмів реплікації баз даних (кінцевої узгодженості достатньо).

Простота розгортання та обслуговування. Відповідно до вимог ТЗ зв'язок побудований на стандартному Ethernet, і ми підтримуємо стандарт Power over Ethernet. Додавання та заміна пристроїв повинні бути швидкими та простими, тому ми подбали про те, щоб можна було попередньо налаштувати пристрої, а потім просто під'єднати їх за потреби. Система повинна бути придатною для використання через десятиліття, тому він повинен залежати лише від компонентів, бібліотек та інструментів, які, ймовірно, залишаться. Ця вимога враховується при розробці апаратного та програмного забезпечення.

2.4 Визначення протоколів зв'язку між сервером і контролером

Протокол зв'язку між сервером і контролером повинен забезпечувати надійність, безпеку, розширюваність і простоту розробки та розгортання, як визначено в ТЗ. Це призвело до наступних рішень. Для того, щоб протокол був простим, але надійним, зв'язок повинен бути бездротовим.

Протокол повинен бути простим. Це включає в себе не тільки низьку складність станів і повідомлень протоколу, але й простоту розбору повідомлень на будь-якому пристрої і будь-якою мовою програмування. Зокрема, розбір повинен бути ефективним на вбудованих пристроях з низькою частотою процесора і пам'яттю. Протокол має бути легко розширюваним. Крім того, повинен бути передбачений механізм для плавного переходу на новішу версію в системі, що працює в режимі реального часу. Це не повинно перешкоджати виконанню вимоги простоти. Крім того, не потрібно покладатися на механізми безпеки, що надаються нижчими рівнями (оскільки вони можуть бути недостатніми або відсутніми). Тому прикладний рівень протоколу повинен забезпечувати достатню автентифікацію і секретність.

Протокол має бути побудований на стандартних, добре відомих технологіях. Код — це відповідальність. Наш код має наші помилки, вимагає специфічних знань і є нашою проблемою. Тому протокол повинен повторно використовувати існуючий код і технології, де це доречно. Необхідно зауважити, що виконання цих вимог зовсім не є тривіальним, оскільки деякі з них, особливо розширюваність проти простоти або безпека проти простоти, можуть в кінцевому підсумку суперечити один одному.

Такі параметри не є суворо обов'язковими, оскільки надійність також може бути досягнута різними способами. Однак, як показано в це є дуже корисним підходом, оскільки дозволяє спростити процедуру навіть перед обличчям вимог, які в іншому випадку потребували б значного ускладнення.

Були розглянуто альтернативи. Початковою пропозицією був протокол, орієнтований на з'єднання, де або сервер, або контролер могли б ініціювати зв'язок. Це дозволило абстрагуватися від таких деталей, як максимальний розмір пакета, обробляти втрачені повторні передачі пакетів, і дозволило б підштовхувати оновлення правил або прошивки з боку сервера. Однак, такий підхід створює безліч проблем, таких як висока вартість відмовостійкості серверів:

— у протоколі зі збереженням стану стан з'єднання потрібно було б або синхронізувати між кількома серверами, що непрактично, або перезапускати весь зв'язок з самого початку у разі збою;

— додаткова надійність лише ціною додаткової складності: знову ж таки через проблеми зі станом сервера;

— більше складнощів на сервері: серверу потрібно буде відстежувати активних контролерів, а також те , хто що сказав і почув;

— зайві процесорні цикли, використання флеш-пам'яті та пам'яті на вбудованих пристроях: хоча ефективна реалізація цих абстракцій може бути непотрібною або може бути написана, неіснуючий код простіше писати, він ефективніший і містить менше помилок.

Визначення та обґрунтування протоколу. Обраний протокол зв'язку є безпровідним, а весь обмін інформацією відбувається у формі "запит контролера → відповідь сервера". Всі запити (включаючи повторні спроби) можуть обслуговуватися незалежно від будь-яких інших минулих, теперішніх або майбутніх запитів від цього або іншого контролера. Це означає, що насправді не існує суворої вимоги обслуговувати всі запити або будь-яку конкретну підмножину запитів одним сервером. Можна використовувати декілька серверів, якщо вони можуть синхронізувати необхідні дані (і навіть для такої синхронізації достатньо кінцевої узгодженості). Таким чином, система може бути налаштована на використання декількох серверів, і очікується, що контролери будуть надсилати запити, включаючи повторні спроби, більш-менш по колу.

Циклічне планування особливо корисне для повторних спроб: якщо виникає проблема з певним сервером або певною частиною мережі, контролер просто повторно надсилає запит на інший сервер, поки не буде отримана хороша відповідь. Таким чином, ймовірність того, що жоден сервер не буде доступний, може бути значно зменшена шляхом розгортання декількох серверів в різних частинах мережі.

За звичайних обставин зв'язок між сервером і контролером не є чутливим до затримок, тому підхід з циклічними повторними спробами не створює проблеми затримок. Тому контролери використовують циклічний підхід з великими таймаутами та експоненціальним відступом, щоб уникнути перевантаження мережі. "Погана" відповідь (наприклад, відповідь, яку неможливо розібрати, або помилка) обробляється так само, як якщо б відповідь не була отримана (за винятком, можливо, інших таймаутів, ведення журналів тощо), тобто повторна спроба надсилається на наступний сервер. Це дозволяє однаково обробляти всі різного роду тимчасові та постійні проблеми з сервером, мережею або іншими ресурсами.

Оновлення системи в реальному часі. Позитивним наслідком незалежності повідомлень і циклічних повторних спроб для всіх помилок є те, що навіть якщо сервер не може проаналізувати запит контролера або контролер не може проаналізувати відповідь сервера, контролер просто повторить спробу з іншим сервером. Тому для переходу на несумісну версію протоколу достатньо розгорнути сервери зі "старим" і "новим" протоколом, і контролери просто повторять спробу, поки не знайдуть сумісний сервер. Разом з тим, що сервер може автоматично доставляти оновлення прошивки, а контролери повідомляють про свою версію прошивки серверу, це робить будь-яке оновлення онлайн-системи тривіальним і повністю автоматичним.

2.5 Обґрунтування вибору мережевого стеку обміну інформацією

Використовується стандартний мережевий стек: Ethernet (IEEE 802.3) як фізичний рівень і рівень передачі даних, IP як мережевий рівень і UDP як транспортний рівень. Для IP підтримуються як IPv4, так і IPv6, а для перетворення адрес між мережею і каналом підтримуються стандартні протоколи ARP або NDP відповідно. IP-адреси можуть бути налаштовані статично або отримані за допомогою DHCP.

UDP проти TCP — стандартна реалізація TCP доступна для всіх пристроїв, які ми будемо використовувати (вона навіть входить до складу ОС реального

часу, що використовується для вбудованих пристроїв). Тому здається, що використання TCP забезпечить переваги без додаткових витрат. Однак, оскільки наш протокол не має стану і є пакетно-орієнтованим, і керує повторними передачами на прикладному рівні, єдиною перевагою TCP буде необмежена довжина "пакету" (на відміну від 64 кБ для UDP [9]), а крім цього, якщо ми вирішимо використовувати його, то в кінцевому підсумку отримаємо сервіс, подібний до UDP, поверх TCP.

Хоча необмежений розмір повідомлень виглядає корисним, насправді він не настільки корисний — єдині повідомлення, які не поміщаються в один UDP-пакет, — це база даних правил і блоки прошивки, і для них ефективніше доставляти їх явними частинами, так що передачу цих великих файлів не потрібно починати спочатку, якщо щось піде не так. Тому, оскільки переваги TCP в цьому випадку не варті накладних витрат на TCP, а обсяг флеш-пам'яті на вбудованих пристроях обмежений, вирішено використовувати UDP.

2.6 Типи повідомлень в системі та зв’язок із контролером

Очікується, що контролери завантажують локальну копію бази даних правил і запитують її замість того, щоб звертатися до сервера щоразу, коли запитується доступ. Вони надсилають журнали доступу, звітують про свій стан і запитують оновлення бази даних правил і прошивки.

Оскільки весь зв'язок повинен ініціюватися контролером, він повинен періодично зв'язуватися з сервером, щоб дізнатися, чи доступна оновлена база даних правил або прошивка.

Всі відповіді мають тег статусу відповіді, значенням якого є одне з OK, ERROR (постійна помилка), TRY_AGAIN (тимчасова помилка). Будь-яка відповідь, що не є OK, повинна розглядатися так, ніби відповідь не надійшла (тобто, зазвичай, необхідна повторна спроба), за винятком можливих різних таймаутів, ведення журналу або планування. Далі показано лише відповіді OK.

Розпізнаються наступні типи повідомлень: PING: keepalive, інформація про версію БД (табл.2.1, табл. 2.2). Звертається до сервера, щоб повідомити про

поточний стан та запросити інформацію про оновлення. Також використовується для налаштування часу контролера.

Таблиця 2.1 — Запит на пінг

Поле	Тип	Опис
time	Ціле число	коли контролер вважає, що час настав
db_version	Ціле число	версія бази даних правил, що використовується в даний час
fw_version	Ціле число	поточна версія прошивки

Таблиця 2.2 — Позитивна відповідь на пінг

Поле	Тип	Опис
time	Ціле число	час сервера
db_version	Ціле число	найновіша доступна версія бази даних правил
fw_version	Ціле число	найновіша доступна версія прошивки

Ведення журналу відповідей через ALOG (табл.2.1) передає журнали доступу та надсилає логи доступу на сервер. Контролери намагаються надсилати журнали доступу якомога швидше, але щоб не втратити їх, вони зберігаються на SD-карті, доки сервер не підтвердить, що їх записано на диск. За потреби журнали можна надсилати кількома партіями.

Таблиця 2.3 — Запит ALOG

Поле	Тип	Опис
records	Масив цілих чисел	Записи
time	Ціле число	Серверний час
card_id	Рядок бітів	Картка, з якої було зроблено запит на доступ
allowed	Логічний	Чи отримано доступ

Відповідь ALOG OK — всі надіслані записи було записано на диск. (Тіло відповіді порожнє).

XFER — передати фрагмент файлу (табл.2.4). Оновлення прошивки та бази правил команда XFER розглядає як непрозорі двійкові блоки. Вони ідентифікуються за типом і версією. Для того, щоб тривіально підтримувати інкрементне завантаження і довільні розміри блоків, контролер явно запитує зміщення і довжину блоку. Одна і та ж версія завжди повинна посилатися на точно ідентичний блок (якщо він існує), навіть якщо запитується з абсолютно незалежного сервера. Сервер може повернути менший блок, але ніколи не довший. Блок довжиною 0 вказує на кінець файлу.

Таблиця 2.4 — Запит XFER

Поле	Тип	Опис
Filetype	enum	Версія DB та FW, що підтримуються
fileversion	Ціле число	Версія файлу
offset	Ціле число	Картка, з якої було зроблено запит на доступ
length	Ціле число	Зсув

Якщо файл не знайдено, сервер поверне помилку TRY_AGAIN. Зазвичай це трапляється через те, що один сервер вже отримав і обробив оновлення, а інший відстає. Контролер просто повторюватиме спроби, доки не знайде готовий сервер. Це єдині відповіді, довші за кілька байт. Сервер надішле відповідь будь-якого розміру, який йому буде запропоновано (до максимального розміру пакета). Кожен контролер несе відповідальність за те, щоб не запитувати фрагменти, які можуть призвести до надто довгих відповідей, які він не в змозі обробити. Це необхідно для забезпечення максимальної ефективності роботи контролерів з різними можливостями.

CRITICAL (табл.2.5) повідомити про критичну проблему. Використовується для повідомлення про критичну проблему, щодо якої сервер повинен вжити негайних заходів.

Таблиця 2.5 —Запит CRITICAL

Поле	Тип	Опис
code	enum	код помилки
message	Текстовий рядок	деталі помилки

КРИТИЧНА ВІДПОВІДЬ ОК: Прийнято, дії вжито (тіло відповіді порожнє).

Єдиними розпізнаними кодами є LOCK_FORCED_OPEN (фізичний замок було відкрито без дозволу) та READER_NOT_RESPONDING (зчитувач не реагує належним чином навіть після багаторазових перезапусків), але припускається, що при підготовці до реального розгортання з'явиться більше застосувань.

Запит доступу (табл.2.6), визначає чи слід надати доступ зараз. Через потенційно високу затримку при обході, у виробництві слід використовувати локальну оцінку, а не запити до сервера. Однак ми включили цей варіант для особливих випадків і як запасний варіант.

Таблиця 2.6 — Запит на доступ

Поле	Тип	Опис
card_id	byte	Картка, яка запросила доступ
allowed	boolean	Отримання доступу

ECHOTEST: відповідь для тестування для тіла запиту. Це корисно для тестування інтеграції. У реальних розгортаннях рекомендується запустити процес, який буде діяти як контролер, що надсилає ECHOTEST (і, можливо, інші) запити і повідомляє про будь-які проблеми (такий процес запускається за замовчуванням).

2.7 Обґрунтування вибору формату пакету

Кодування записів. Всі запити та відповіді, а також зовнішній конверт пакета є "записами", тобто невеликими відображеннями ключ-значення з фіксованими іменами та типами ключів. Тому спочатку ми хотіли просто передавати "структури C" (тобто двійкові блоки з фіксованими зміщеннями полів) і жорстко кодувати зміщення полів у прошивці сервера та контролера. Однак такий підхід має багато недоліків.

Кодування записів, всі запити та відповіді, а також зовнішній конверт пакета є "записами", тобто невеликими відображеннями ключ-значення з фіксованими іменами та типами ключів. Тому спочатку ми хотіли просто передавати "структури C" (тобто двійкові блоки з фіксованими зміщеннями полів) і жорстко кодувати зміщення полів у прошивці сервера та контролера.

Однак такий підхід має багато недоліків:

— будь-яке розширення буде несумісною зміною, а отже, вимагатиме повну процедуру оновлення, хоча ця процедура проста, коли вона запущена, система вимагає більше серверів для досягнення того ж рівня надмірності; і це може змусити адміністраторів нервувати;

— ми можемо неправильно розібрати пакет, навіть не помітивши цього, якщо довжина збігається, коли довжина не збігається, ми не знаємо нічого більш конкретного, ніж "синтаксичний розбір не вдалося";

— блок не самоописується, і тому про нього нічого не відомо без контексту зовнішнього конверта, в якому вказано версію та опис полів для цієї версії.

Особливо занепокоєння щодо помилок синтаксичного аналізу є досить значним, щоб виправдати самоописуюче кодування. Тому нам потрібне кодування з наступними властивостями:

— самоопис — імена та типи ключів повинні бути присутніми в закодованих даних;

— виразним — має бути можливість включати всі необхідні типи та довільно вкладати їх у вигляді масивів або підзаписів; мають підтримуватися необов'язкові поля;

— бінарно-безпечні — здатні передавати довільні двійкові дані (наприклад, ідентифікатори карток або фрагменти файлів) без необхідності додаткового кодування;

— не несумісні за замовчуванням — коли вносяться зворотньо-сумісні зміни (наприклад, додавання нового необов'язкового поля або видалення поля, яке було необов'язковим), старий і новий код повинні мати можливість взаємодіяти без змін;

— підходять для вбудованих пристроїв — кодування та декодування має бути швидким, з використанням невеликого розміру коду та створенням невеликих повідомлень;

— стандартний, з існуючими бібліотеками — коли код проблема розробника, чим менше коду пишеться, тим менше коду потрібно буде підтримувати в майбутньому.

Цим вимогам чудово відповідає стислий двійковий опис об'єктів CBOR (Concise Binary Object Representation) [3] — формат даних, розроблений для зв'язку з обмеженими вузлами. Якщо зустрічається дублікат тегу, це вважається помилкою. На додаток до перевірки адекватності, це може запобігти деяким атакам, пов'язаним з переповненням. Для представлення записів ми використовуємо масиви семантично позначених елементів CBOR (ці еквівалентні масивам пар (тег, дані), де дані строго типізовані).

Невідомі теги ігноруються, і з точки зору синтаксичного аналізу всі поля є необов'язковими. Таким чином, єдине, що сервер і контролер повинні мати спільного для спілкування — це інтерпретації тегів (що має сенс, якщо вони хочуть використовувати значення для чогось корисного).

Запити, відповіді. Для всіх запитів і відповідей запис, як зазначено вище, тегується семантичним тегом для відповідного типу повідомлення, а у випадку записів відповідей тег, у свою чергу, тегується статусом відповіді.

"Конверт" - версія, адресація, шифрування. Зовнішній рівень повідомлень (спільний для запитів і відповідей) забезпечує адресацію і шифрування. Це запис з полями, як зазначено в таблиці 3.10. До закодованого запису додається 4-байтовий "магічний номер", що містить байти [68, 69, 65, 68] ('DEAD' в ASCII), який ідентифікує це повідомлення як повідомлення про тупик (Deadlock).

Ідентифікатор версії Пакет слід вважати недійсним, якщо він не відповідає відомій версії. Це необхідно для підтримки оновлень системи у реальному часі.

Ідентифікатор контролера — унікальний ідентифікатор відправника або передбачуваного одержувача. Слугує для адресації. Включення форми адресації на прикладному рівні відокремлює "логічну" адресацію від "фізичної" (тобто мережевої), що дозволяє Deadlock функціонувати через NAT, з широкомовними/багатоадресними/багатоадресними IP-адресами тощо.

Nonce — випадково згенеровані байти. Відповідає відповіді на запит: якщо Nonce запиту дорівнює x, то Nonce відповідної відповіді має бути x $\oplus$ 1.

Корисне навантаження Запит/відповідь, зашифрований відповідно до ТЗ. Зашифрований ключем для даного контролера з використанням nonce. Максимальний розмір повідомлення (у закодованому та зашифрованому виді).

2.8 Визначення параметрів безпеки системи

Хоча бібліотеки, що реалізують криптографічні примітиви, існують, вони, як правило, не роблять захист програми особливо простим: розробник повинен знати, що саме потребує захисту; які примітиви (такі як шифр, режим шифрування, контрольні суми, підписи) підходять для конкретного випадку використання, що вони обіцяють, які їхні слабкі сторони і чи є вони проблемою в даному випадку використання; він повинен враховувати потенційні атаки з побічних каналів, атаки з відтворенням , тощо; і він повинен переконатися, що

інші розробники знають про всі ці міркування. Як показують численні звіти про вразливості, що публікуються щомісяця, це нелегке завдання.

Якщо не замикати комп'ютер у комірчині без електрики, то найкращий спосіб захистити систему - це довірити її експерту. На щастя, у 2013 році було опубліковано специфікацію інтерфейсу бібліотеки NaCl [7] та декілька реалізацій, з метою надання розробникам простого криптоінструментарію "за замовчуванням". В роботі [2] визначено особливості такої бібліотеки.

Система розробляється для роботи через ненадійні мережі, і повинна протистояти як пасивним, так і активним атакам. Тому шифруються та автентифікуються всі повідомлення від/до заданого контролера симетричним ключем, специфічним для пристрою, використовуючи функцію NaCl secret_box(nonce, ключ, корисне навантаження), яка визначає секретність та цілісність за умови, що nonce не використовується більше одного разу [7]. Також Nonce визначається шляхом генерації, де у кожному пакеті використовується 24 випадкових байти, що забезпечує незначну ймовірність колізій (швидке наближення парадоксу дня народження говорить, що ймовірність досягає 50% після більш ніж 10 пакетів, що є дуже багато).

Симетрична криптографія була обрана з міркувань продуктивності, але як тільки буде готове апаратне та програмне забезпечення контролера, ми плануємо провести бенчмарки і перейти на асиметричну криптографію, якщо це можливо, щоб уникнути необхідності копіювати секрет в декілька місць. За замовчуванням примітивами NaCl є потоковий шифр Salsa20 для симетричного шифрування та MAC Poly1305 для автентифікації повідомлень. Як детально описано в [11], вони є безпечними і продуктивними, не залежать від будь-якої форми апаратного прискорення, що добре відповідає нашим вимогам.

Гарантії безпеки за умови, що nonce не використовується більше одного разу, secret_box(nonce, key, payload):

— секретність, неможливо розшифрувати повідомлення без знання ключа;

— цілісність, якщо повідомлення успішно розшифровано, то не може бути випадкової або цілеспрямованої модифікації Nonce або зашифрованого корисного навантаження третьою стороною;

— стійкість до атак н а синхронізацію: реалізації намагаються завжди виконувати однаковий обсяг роботи.

Крім того, ідемпотентність протоколу та використання Nonce запобігає атакам повтору: якщо зловмисник спробує повторити запит до сервера, нічого поганого не станеться, оскільки всі запити є ідемпотентними; якщо він повторить відповідь контролеру, його Nonce не збігатиметься з жодною з відповідей, які контролер очікує в даний момент, а тому він проігнорує фальшиву відповідь. "Випадковий" в даному випадку не означає криптографічно захищену випадковість — Nonce можуть бути передбачуваними (вони все одно надсилаються відкритим текстом разом з корисним навантаженням), єдиною вимогою є рівномірний розподіл, щоб забезпечити низьку ймовірність колізій. Той факт, що NaCl не потребує джерела хорошої випадковості, у вбудованих середовищах дуже вітається. вигляді) становить 63 кБ (для того, щоб зручно поміститися в UDP- пакеті).

2.9 Визначення правил доступу в системі

Ключове зауваження при розробці правил доступу полягає в наступному. Загальність досягається за рахунок складності. Для будь-якої програми більшість правил виглядатимуть однаково, і тому, якщо правила є загальними, вони будуть невиправдано складними в типовому випадку використання. Здебільшого користувача дратуватиме необхідність щоразу вводити схожі правила, замість того, щоб користуватися загальними правилами.

Через цю проблему вирішено створити два різних рівні правил доступу: низькорівневий рівень, який є загальним і простим, і високорівневий рівень, специфічний для конкретного домену, який оптимізовано для типового використання в даному домені. Високорівневий рівень ґрунтується на примітивах, наданих низькорівневим рівнем, і для різних випадків використання

(наприклад, для кампусів і готелів) слід розробляти різні високорівневі правила. Це забезпечує гнучкість і зручність одночасно.

Для того, щоб підтримувати як типові випадки використання, так і унікальні сніжинки в одній інсталяції, високорівневі реалізації правил не повинні припускати нічого про правила, встановлені в системі - їм не дозволяється необережно видаляти існуючі правила або припускати, що існують тільки ті правила, про які вони знають. Вони повинні відображати низькорівневе представлення для правил, які вони не можуть інтерпретувати у своїй високорівневій моделі.

Щоб полегшити цю взаємодію між правилами високого і низького рівнів і забезпечити узгодженість, ми придумали поняття набору правил: кожне правило в оновлювати або видаляти лише цілі набори правил, а не окремі правила. Операції над наборами правил є атомарними. Додаток, що реалізує правила високого рівня, повинен працювати тільки з наборами правил, створеними цим додатком.

Існує механізм забезпечення дотримання цього обмеження. Низькорівневі, внутрішні правила повинні бути достатньо загальними, щоб підтримувати будь-який варіант використання, але їх легко компілювати як комп'ютерам, так і людям.

Формат внутрішніх правил. Для того, щоб охопити всі можливі випадки використання, найпростіший підхід полягає в тому, щоб дозволити задавати правила доступу у вигляді будь-якої булевої формули над ідентичностями, точками доступу та часовими характеристиками. Однак, це призводить до наступних проблем:

— людині важко швидко міркувати про результат того чи іншого запиту;

— необхідна повна оцінка кожного запиту, що може зайняти багато пам'яті або часу; непрактично проводити попередні обчислення для великих вхідних даних;

— для нормальної функціональності логіка оцінки та всі дані, необхідні для оцінки, повинні бути вбудовані в контролери, що порушує принцип проектування "зробити вбудовані пристрої простими";

— невелика зміна вхідних даних або формул може мати як завгодно великі наслідки, що заважає як спробам зрозуміти, чому щось відбувається, так і попереднім обчисленням.

Щоб уникнути цих проблем, було обрано наступну модель.

Кожна точка доступу має лише один тип; для кожного типу можна додати правила, які відповідають часовим характеристикам та ідентифікаційним виразом для відповіді "Дозволити" або "Заборонити". Правила строго впорядковані за пріоритетом. Це реалізовано в обраній нами СУБД PostgreSQL за допомогою механізму безпеки на рівні рядків [11]. Процес оцінювання, як показано на рисунку 4.1, виглядає наступним чином:

1) знайдіть тип цієї точки доступу, виберіть її правила;

2) виберіть правила з відповідною специфікацією часу;

3) виберіть правила, в яких ця тотожність збігається з виразом тотожності правила;

4) виберіть (єдине) правило з найвищим пріоритетом.

При цьому вибирається єдине правило, яке однозначно дозволяє або забороняє доступ.

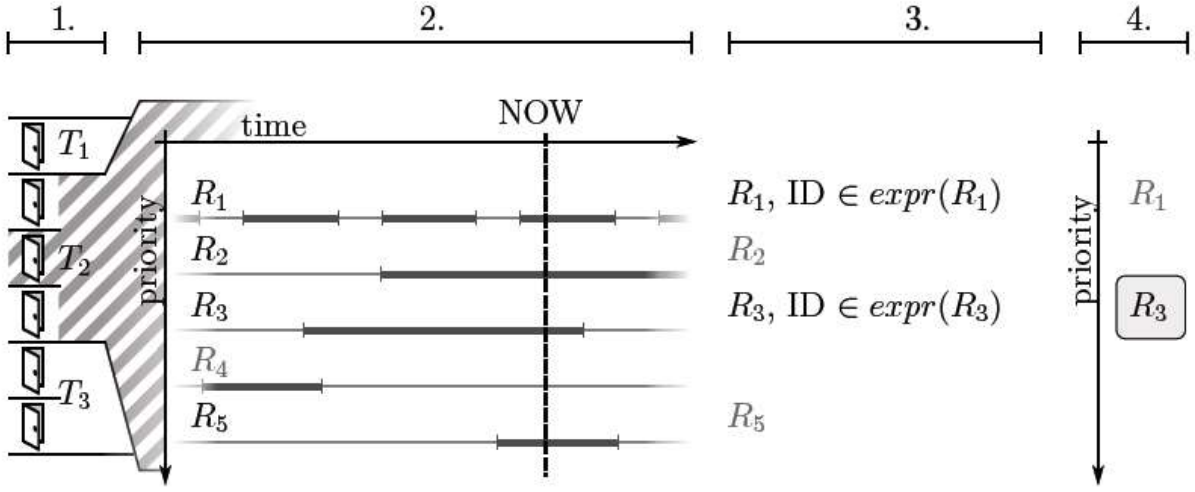


Рисунок 2.1 — Потік оцінки правил

Вирази ідентичності, вираз тотожності — це (обмежений) булевий фор- мул лише над тотожностями, який реалізує узагальнення контролю доступу за групами. Реалізація загальних булевих формул (наприклад, за допомогою операторів AND, OR та NOT) була б можливою, але для підтримки NOT нам довелося б або зберігати доповнення, що може вимагати багато пам'яті для невеликих вхідних даних, або зробити обчислення менш простим, що суперечить принципу простоти контролерів. Тому у виразах тотожності використовуються оператори INCLUDE та EXCLUDE, які еквівалентні об'єднанню та різниці множин. Вони еквівалентні (навіть за складністю виразу) загальним булевим формулам, якщо задано набір "цікавих" тотожностей (які так і є, оскільки "будь-який ідентифікатор, окрім цього" не є корисним правилом).

Тому визначають вирази ідентичності як:

$$\text{expr} ::= \text{INCLUDE } x_1, \ldots, \text{INCLUDE } x_m, \text{EXCLUDE } x_{m+1}, \ldots, \text{EXCLUDE } x_n$$

де $x ::= \text{expr} \mid \text{identity}$, з семантикою "об'єднання всіх включених підвиразів мінус об'єднання всіх виключених підвиразів".

Обґрунтування такого поділу — розділення оцінювання правил на вирази тотожності та вирази "час+місце" означає, що правила легше оцінювати: людина (або комп'ютер) може оцінювати ці два вирази незалежно, і на питання "чому так відбувається" легше відповісти. Крім того, таким чином легше знаходити класи еквівалентності на входах, оскільки в цій моделі одне правило часу+місця відповідає одному виразу тотожності, а не довільним комбінаціям. Це робить практичним попереднє обчислення деяких правил та їх повторне обчислення поетапно при змінах.

Запит швидкого доступу: відношення "у виразі". Як правило, правила будуть часто запитуватися (особливо при створенні локальних баз даних правил для контролерів) і нечасто змінюватися. Тому ми можемо заощадити роботу і час, попередньо обчисливши деяку інформацію. Наразі ми припускаємо, що в типовому розгортанні буде небагато правил і багаторівневих виразів

ідентичності. Тому ми попередньо обчислюємо відношення "у виразі ідентичності" - для кожної ідентичності (тобто для всіх листків виразів) ми піднімаємося по дереву виразів (або, фактично, DAG) і зберігаємо кортеж (ідентичність, вираз), коли ідентичність включається у вираз (з урахуванням операцій INCLUDE/EXCLUDE). Оскільки вирази є ациклічними, щоразу, коли нам потрібно включити/виключити під-вираз, ми можемо повторно обчислити вирази в порядку залежностей (а отже, рівно один раз).

Для того, щоб вибрати правило, застосовне для даного запиту доступу згідно з ТЗ на кроці 3 просто вибираються правила, в яких існує кортеж (ідентичність, вираз). Аналогічно, при створенні локальної бази даних для контролерів використовується тільки сплющене відношення, а не оригінальна ієрархія.

Коли вираз тотожності змінюється, легко інкрементно переобчислити лише ті частини, які зазнали змін: ми просто шукаємо в DAG, переобчислюючи вузли під час їх відвідування.

Локальна копія бази даних на контролерах будується на цьому дворівневому підході, що передбачає розмежування ідентифікаційних виразів та часових специфікацій. Зауважте, що будь-який контролер обслуговує одну точку доступу, і тому про частину правил "де" вже подбано - кожна точка доступу знає лише про правила, що належать до її типу.

Сервер прослуховує повідомлення про "зміну правил" з бази даних і перестворює локальні бази даних, специфічні для контролера, коли це необхідно. Конкретний формат локальних баз даних виходить за рамки цієї тези.

Інтеграція з існуючими системами. Відповідно до вимог ТЗ, дані можна імпортувати з інших систем і прозоро "вставляти" їх у правила доступу. Це робиться за допомогою програми, яка генерує плоскі вирази ідентичності виду:

X .= [INCLUDE person , INCLUDE person, ...],

для кожної групи X, яку потрібно імпортувати. Ці групи позначені і вважаються примітивами, і вони можуть бути змінені лише шляхом створення групи Y, яка

включає X і далі включає або виключає те, що потрібно скоригувати при імпорті. Таким чином, при зміні базових даних, "латки" не будуть порушені.

3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ СЕРВЕРНОЇ ЧАСТИНИ СИСТЕМИ

В епоху цифрових інновацій, питання безпеки та оптимізованого управління доступом стає все більш важливим. Системи дистанційного керування доступом дозволяють забезпечувати високий рівень захисту, комфорт використання та гнучкість у адмініструванні доступу до активів організації. Цей ініціатива має на меті розробити систему віддаленого керування доступом, яка включає розробку веб-сервера та програми адміністратора. Основною ціллю цієї роботи є розробка продуктивної та безпечної системи віддаленого керування доступом, яка забезпечить адміністрування доступу користувачів до різних активів через веб-інтерфейс. Завдання включають дослідження можливих засобів реалізації, вибір відповідної технологічної платформи, розробку веб-сервера та програми адміністратора, а також тестування та налагодження системи.

3.1 Організація даних для обміну даними в системі

Реплікація та обхід відмови сервера повинні бути простими, а отже, будь-які дані/стани, які потрібні серверу, повинні легко реплікуватися. Тому сервер може залежати лише від вмісту реляційної бази даних (для якої існують механізми реплікації) - він може кешувати або попередньо обчислювати деякі значення, але в іншому випадку весь вивід повинен бути чистою функцією вмісту бази даних. У наступних підрозділах наведено огляд даних, з якими працює сервер.

Управління точками доступу та журнали доступу. Для кожної точки доступу зберігається її тип, необов'язковий опис і те, який контролер до неї підключено. Журнали доступу (з даними, зазначеними в повідомленні, відповідно до ТЗ) зберігаються в базі даних. Для того, щоб виконати гарантію ідемпотентності протоколу, зберігаються тільки журнали з унікальною комбінацією атрибутів.

Дані про стан контролерів, зокрема час останнього PING, версія бази правил, версія прошивки та місцевий час (для вимірювання дрейфу) СЕРВЕРА зберігаються. Оскільки журнал бази даних записується на диск під час звертаня, журнали зберігаються протягом тривалого часу до того моменту, коли ми надсилаємо відповідь "ОК" контролеру.

Інші частини системи, такі як моніторинг або інтерфейс керування, можуть використовувати цю інформацію на власний розсуд (наприклад, для попередження, якщо контролер занадто довго мовчав або якщо його дані застаріли).

Специфікація правил доступу та управління ідентичностями. Ідентичності та вирази утворюють групу DAG, як описано в попередньому розділі. Представляється це шляхом зберігання ребер (позначених операцією INCLUDE або EXCLUDE) і вузлів (що містять посилання на ідентичність або під-вираз) у базі даних. Крім того, обчислюються відношення "у виразі", як описано в попередньому розділі.

Правила доступу зберігаються у вигляді відношення на основі типів PoA, виразів ідентифікації, часових специфікацій, пріоритету та {, }. Специфікації часу можуть містити список днів тижня, діапазон дат і час доби, і вони функціонують як маски - якщо їх не вказано, вони відповідають будь-якому дню тижня/даті/часу. Операції (такі як об'єднання/перетин) над цими масками не підтримуються - замість цього слід створити кілька правил з відповідними пріоритетами.

3.2 Основні компоненти серверної частини

Функціональність сервера була розділена на 3 окремі компоненти з мінімалістичними інтерфейсами. Вони запускаються як окремі процеси, і не передбачається, що вони працюють на одній машині.

Інтерфейс "спільних файлів". Щоб уникнути тісного зв'язку між цими модулями, зазвичай єдиним спільним інтерфейсом між ними (окрім спільної бази даних) є файлова система: під час конфігурування розгортання надається каталог

спільної файлової системи з доступом на читання та запис, і компоненти взаємодіють, створюючи файли в цьому місці та отримуючи до них доступ. Зазвичай цього достатньо, оскільки компоненти розроблені так, щоб працювати незалежно, що було створено іншими компонентами. Зокрема, метою декількох компонентів є створення СЕРВЕРА. Файли, призначені для передачі на контролери за допомогою повідомлення XFER, і для них ми створили просту загальну бібліотеку для відкриття файлів, призначених для конкретного контролера (за бажанням, з можливістю переходу до файлів, спільних для всіх контролерів).

Deadserver: взаємодіє з контролерами. Вислуховує повторні запити контролерів на UDP-сокеті та надсилає відповіді згідно з протоколом. Для запитів PING та XFER шукає файли за допомогою механізму Інтерфейсу "спільних файлів".

Deadapi: API для зовнішнього світу. Надає HTTP API, що використовується веб-інтерфейсом управління та моніторингу, а також інтерфейс командного рядка. Таким чином, з'єднує зовнішній світ і базу даних за допомогою простого CRUD REST API.

Він підтримує підштовхування подій через потокову довготривалу HTTP-відповідь, відповідно до специфікації Server-sent events/Event source [6]. Події запускаються через механізм LISTEN/NOTIFY pub/sub у Postgres [10], а база даних, в свою чергу, містить тригери, які надсилають NOTIFY на певні зміни рядків таблиці. Таким чином, зміни даних можуть передаватися клієнтам, які можуть використовувати стандартний API джерела подій для підписки на них.

Забезпечує швидкий спосіб оновлення прошивки: можна завантажити образ прошивки разом зі списком ідентифікаторів контролерів, a deadapi просто вивантажить (або зв'яже) файл у підтеки, призначені для даних контролерів .

Deadaux: допоміжні завдання, що підтримують інші компоненти. deadaux — це набір допоміжних модулів, які підтримують функціональність deadserver та deadapi, а також дуже простий диспетчер, який запускає модулі в окремих потоках. За замовчуванням до складу deadaux входять наступні модулі:

Offlinedb: Основним обов'язком цього модуля є створення копії бази даних правил, яку контролери використовують для локальної оцінки роботи. Його основний потік використовує механізм pub/sub у Postgres СЕРВЕЛРІЗАТЕН/NOTIFY [10], щоб отримувати сповіщення СЕРВЕРА при зміні правил. При зміні він генерує нові версії файлів і скидає їх туди, де контролери можуть їх знайти за допомогою механізму загальних файлів через Інтерфейс "спільних файлів".

Echotest: Використовує клієнтську бібліотеку контролера для періодичного надсилання меседжів ECHOTEST на сервер і перевірки відповіді. Може бути налаштовано, наприклад, на надсилання електронного листа у разі виникнення проблем.

3.3 Вибір мови програмування та технологій для веб-сервера

Для розробки веб-сервера було обрано: Spring Boot (Java).

Переваги: висока надійність, багатий набір функцій для розробки корпоративних додатків, хороша підтримка спільноти.

Низька вартість входу для нових розробників: більшість потенційних майбутніх розробників вже знайомі з мовою.

Відмінне співвідношення простоти і можливостей.

Ефективні конструкції, що сприяють гарному дизайну та правильному коду: Конструкції для орієнтації, такі як декоратори, заохочують модульність та композицію, а також наприклад, контекстні менеджери допомагають забезпечити правильне управління ресурсами, транзакціями тощо.

Доступні хороші бібліотеки: бібліотеки для загальних завдань, таких як взаємодія з БД, обслуговування UDP або HTTP запитів і багато іншого, є легкодоступними, добре відомими і добре протестованими.

Недоліки: відносно складна конфігурація, висока вимогливість до ресурсів.

Основні компоненти веб-сервера:

Контролери – обробляють HTTP-запити та відповідають за взаємодію з клієнтом.

Сервіси – реалізують бізнес-логіку додатку. Репозиторії – відповідають за доступ до бази даних.

Моделі – описують структуру даних, що використовуються у додатку.

Для СУБД потрібна версія PostgreSQL >= 9.3. Ми використовуємо нестандартні можливості Postgres, такі як процедурна мова PL/pgSQL в базі даних [8] для попереднього обчислення правил, або система NOTIFY/LISTEN pub/sub для сповіщення deadaux про зміни правил доступу.

Деякі з використаних бібліотек (принаймні psycopg та pynacl) використовують власні прив'язки, а тому працюють лише з CPython для користувачів до різних ресурсів через веб-інтерфейс. Завдання включають аналіз можливих засобів реалізації, вибір відповідної технологічної платформи, розробку веб-сервера та додатку адміністратора, а також тестування та налагодження системи.

Опис концепції втілення. В основу концепції ідентифікації користувачів при доступі було покладено створення унікальних ключів (UUID), які забезпечують високий рівень захисту та унікальність кожного користувача. UUID (Universally Unique Identifier) – це 128-бітове число, що використовується для унікальної ідентифікації даних в комп’ютерних системах.

Аргументи вибору UUID:

Унікальність. Кожен згенерований UUID є унікальним, що виключає можливість дублювання ідентифікаторів користувачів.

Безпека. Високий рівень ентропії та складність UUID робить його практично неможливим для підробки або випадкового збігу.

Стандартизація. UUID є стандартом, який широко використовується у багатьох системах та додатках, що забезпечує сумісність та легкість інтеграції.

Простота генерації. Багато мов програмування та фреймворків мають вбудовані бібліотеки для генерації UUID, що спрощує процес їх створення та використання.

Формат UUID складається з 32 шістнадцяткових цифр, розділених на п’ять груп, що відображається у вигляді рядка, наприклад: fe73090a-ab57-4cc4-b21e-cef660ba2be8. Цей формат складається з: 8 шістнадцяткових цифр, 4 шістнадцяткових цифри, 4 шістнадцяткових цифри, 4 шістнадцяткових цифри, 12 шістнадцяткових цифр

Процес ідентифікації полягає в наступному:

Генерація UUID: Кожному користувачу при створенні його облікового запису присвоюється унікальний UUID, що зберігається в базі даних.

Аутентифікація: При спробі доступу користувач пред'являє свій UUID, який перевіряється на відповідність запису в базі даних.

Авторизація: Після успішної аутентифікації система визначає рівень доступу користувача до ресурсів, заснований на його UUID та пов'язаних з ним правами доступу.

Переваги використання UUID:

Масштабованість: Система з UUID легко масштабується, оскільки унікальність ідентифікаторів гарантує відсутність конфліктів при збільшенні кількості користувачів.

Незалежність: UUID є незалежними від контексту системи, що дозволяє використовувати їх у різних середовищах без необхідності змінювати формат або алгоритм генерації.

Інтеграція: Завдяки стандартизованому формату UUID легко інтегруються з іншими системами та сервісами, що використовують цей формат для ідентифікації.

Вибір СУБД та опис структури бази даних. PostgreSQL – це потужна, відкрита реляційна система управління базами даних (РСУБД), яка широко використовується завдяки своїй надійності, масштабованості та багатим функціональним можливостям. Основні характеристики PostgreSQL включають:

Надійність та стійкість. PostgreSQL забезпечує високу надійність завдяки механізмам резервного копіювання та відновлення, що є критично важливим для систем безпеки.

Підтримка складних запитів. Завдяки потужному механізму оптимізації запитів PostgreSQL дозволяє ефективно обробляти складні та ресурсомісткі запити.

Безпека: PostgreSQL має вбудовані механізми контролю доступу та аутентифікації, що важливо для системи, яка керує доступом користувачів.

Масштабованість. Система здатна ефективно масштабуватися разом з ростом кількості користувачів та обсягів даних, що робить її придатною для довгострокового використання.

Розширюваність. Можливість створення кастомних розширень та модулів дозволяє адаптувати PostgreSQL під специфічні вимоги проекту.

3.4 Архітектура та структура бази даних

Як пояснювалося в розділі 3.1, структура бази даних — це повна інформація (за винятком кешування/попередніх обчислень) про дані, з якими працює Deadlock.

База даних містить наступні складові.

Таблиці користувачів: Зберігають інформацію про користувачів, включаючи унікальні ідентифікатори (UUID), імена, ролі та права доступу.

Таблиці доступу: Містять дані про спроби доступу, включаючи час, місце, статус доступу.

Індекси: Використовуються для прискорення запитів, особливо по полях, що часто використовуються у фільтрації та сортуванні (наприклад, UUID, час доступу).

Реалізація взаємодії з базою даних через ORM (Object-Relational Mapping): Використання ORM (наприклад, Hibernate для Java) дозволяє ефективно управляти базою даних на рівні об'єктів, що спрощує розробку та підтримку коду.

Транзакції: Застосування транзакцій забезпечує цілісність даних та дозволяє безпечно виконувати послідовність взаємопов'язаних операцій.

Реплікація та резервне копіювання: Налаштування механізмів реплікації та регулярного резервного копіювання забезпечують надійність та безперервність роботи системи. На рисунку 3.1 показано діаграму "сутність зв'язок" для базової схеми бази даних.

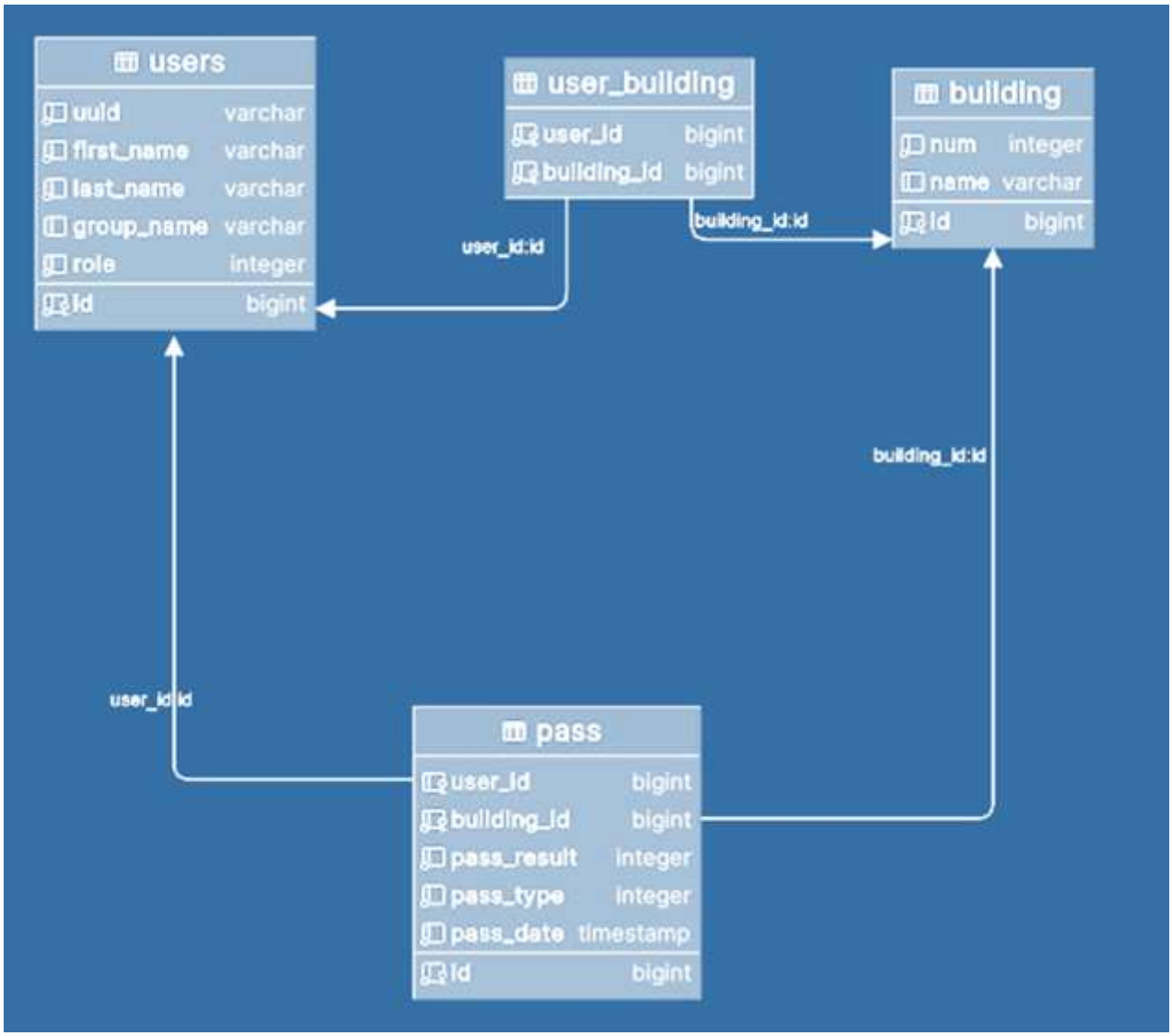


Рисунок 3.1 — Діаграма "сутність-зв'язок" для схеми бази даних

На додаток до вищезазначеного, на основі даних обчислюється відношення "у виразі", як описано в розділі 2.3.

Для економії часу та ресурсів ці обчислення реалізовано в базі даних за допомогою PL/pgSQL [8].

Функція переобчислення запускається змінами в таблиці in_expr_edge за допомогою механізму CREATE TRIGGER в SQL. При зміні вона рекурсивно обходить вираз ідентичності DAG, позначаючи те, що потрібно переобчислити в допоміжному

Таблиця _mr_recalculate. Перерахунок відбувається всередині транзакції, тому інші запити не можуть бачити часткові зміни в таблиці in_expr - цілісність під час перерахунку забезпечено.

Абстракція CryptoBox. Щоб уникнути випадкового розкриття секретних ключів (наприклад, як частина зареєстрованого трасування) і забезпечити хорошу абстракцію використовуваної криптографії, ми створили інтерфейс CryptoBox: чорний ящик, який може виконувати шифрування і дешифрування для конкретного контролера. Таким чином, ми уникаємо прямої передачі секретного ключа, тим самим зменшуючи ризик того, що він потрапить туди, куди не повинен. (Звичайно, ми не можемо повністю приховати його від процесу, оскільки за звичайних обставин він має бути відображений у тому ж адресному просторі, але ми можемо принаймні уникнути його розкриття без помітних зусиль). Це також абстрагується від специфіки конкретних криптографічних примітивів, що використовуються, що дозволяє переключитися на інший метод (наприклад, на асиметричну криптографію) без необхідності змінювати будь-який з код за допомогою CryptoBox.

Цей API натхненний до бібліотеки NaCl [12]. Ви "версія файлу означає вміст файлу". Протокол гарантує, що певна версія файлу завжди вказує на той самий вміст, до останнього байта. Для того, щоб мати можливість це має гарантувати:

— завжди записувати у тимчасовий файл і перейменовувати його на розпізнане ім'я лише після того, як він буде готовий, покладаючись на атомарність виклику перейменування POSIX у тій самій файловій системі [14];

— виводиться ім'я файлу з його вмісту: обчислюємо 32-бітний хеш FNV-1a [5] під час запису файлу і використовуємо його як версію, хеш-

алгоритм FNV-1a було обрано через низьку ймовірність колізій у 32-бітному варіанті та хорошу продуктивність, особливо на довгих вхідних даних.

3.5 Опис роботи та основні функції веб-серверу

Веб-сервер забезпечує обробку запитів, пов'язаних із пропуском користувачів у та з будівель. Він містить два основні ендпоінти, які дозволяють фіксувати вхід і вихід користувачів, використовуючи їхні унікальні ідентифікатори (UUID) та номери будівель.

Прийом HTTP-запитів: Контролер приймає HTTP-запити від клієнтів, наприклад, веб-додатків або мобільних додатків), обробляє їх та викликає відповідні сервіси для виконання необхідних дій.

Обробка параметрів запитів: Запити містять параметри, які сервер витягує та використовує для подальшої обробки. У випадку даного сервера, параметри включають UUID користувача та номер будівлі.

Опис ендпоінтів.

Ендпоінт для входу (/pass/in):

Тип запиту: POST

Опис: Цей ендпоінт використовується для фіксації входу користувача в будівлю.

Параметри:

uuid (рядок): Унікальний ідентифікатор користувача.

buildingNumber (ціле число): Номер будівлі, в яку користувач входить.

Процес: При отриманні запиту контролер передає UUID користувача та номер будівлі у сервісний метод, який реєструє факт входу. Після успішної реєстрації сервер повертає відповідь з кодом статусу 200 (OK).

Ендпоінт для виходу (/pass/out):

Тип запиту: POST

Опис: Цей ендпоінт використовується для фіксації виходу користувача з будівлі.

Параметри:

uuid (рядок): Унікальний ідентифікатор користувача.

buildingNumber (ціле число): Номер будівлі, з якої користувач виходить.

Відповіді сервера. У випадку успішного виконання обох ендпоінтів, сервер повертає відповідь з кодом статусу 200 (OK) без додаткового контенту у тілі відповіді.

У випадку виникнення помилки (у користувача немає доступу до корпусу, користувача з таким uuid не існує), сервер поверне відповідь з кодом 400 (Bad request) та описом проблеми у тілі.

3.7 Опис роботи веб-застосунку адміністратора

Веб-застосунок є закритим від сторонніх користувачів тому на початку буде просити авторизуватись за допомогою логіна та пароля (рис. 3.2).

Рисунок 3.2 — Вікно авторизації адміністратора

Після авторизації адміністратор потрапить на сторінку з історією всіх пропусків відсортованій по даті.

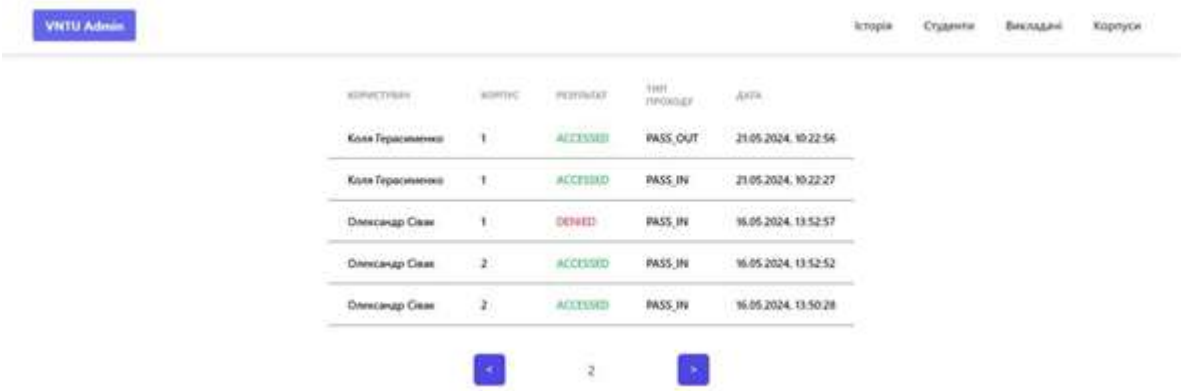


Рисунок 3.3 — Вікно ведення журналу користувачів та часу доступу

Також доступні такі сторінки як: Студенти, Викладачі, Корпуси.

На сторінці студенти доступні для перегляду всі студенти в системі (рис.3.4):

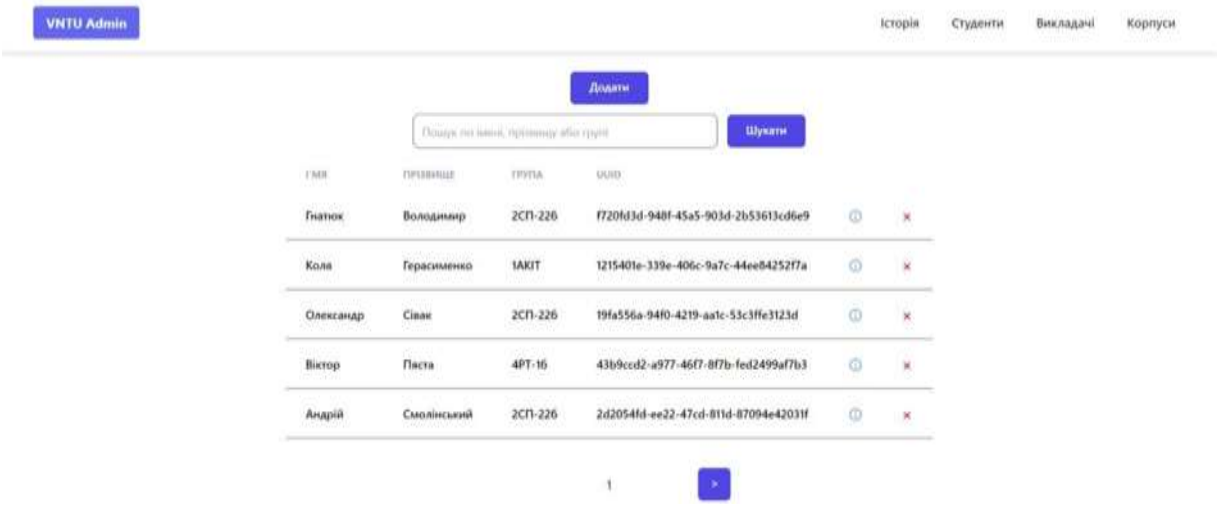


Рисунок 3.4 — Вікно ведення журналу користувачів та часу доступу

Крім того, система реалізує доступний розумний пошук та виклик модального вікна для додавання нового студента:

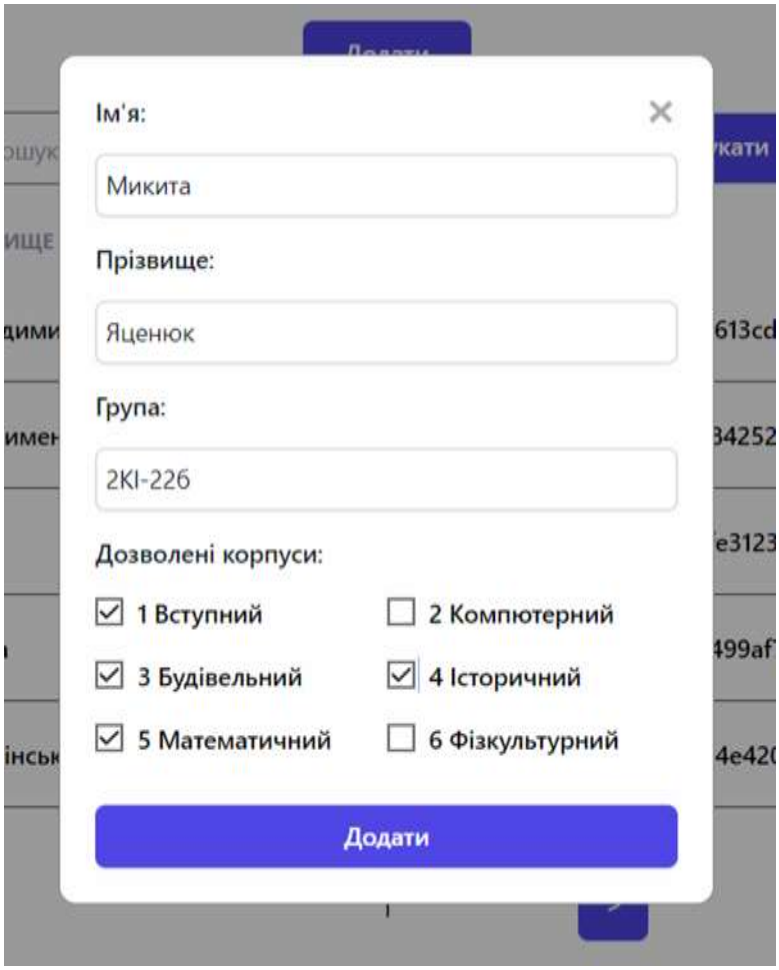


Рисунок 3.5 — Вікно призначення дозволів користувачу

У відповідь на додавання отримується повідомлення:

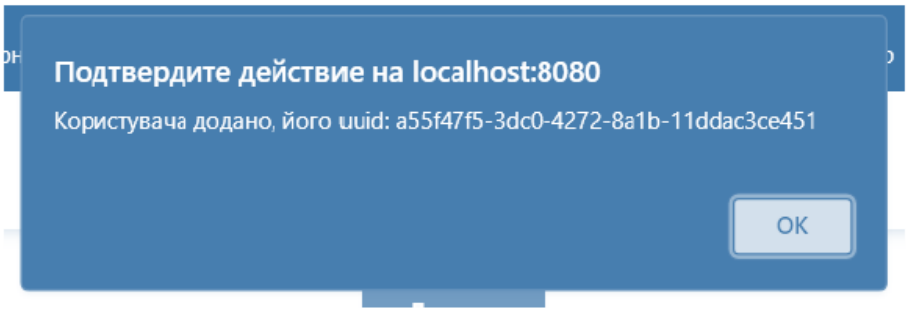


Рисунок 3.6 — Вікно призначення дозволів користувачу

В результаті у системі буде створений користувач з відповідними правами.

На персональній сторінці ми можемо отримати інформацію по цьому користувачу.

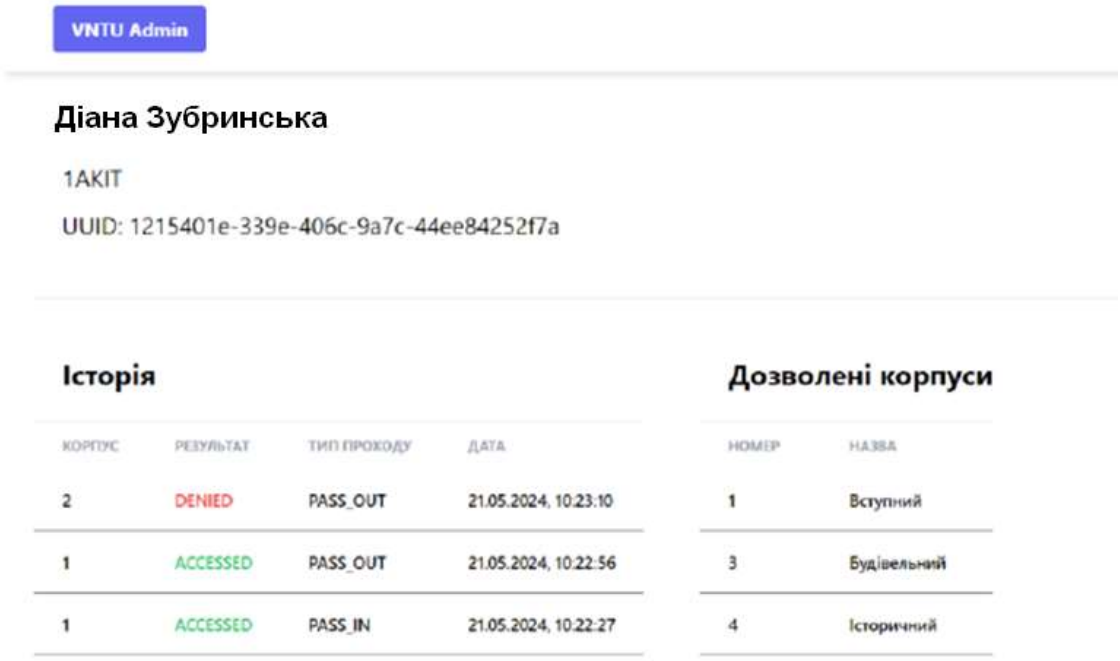


Рисунок 3.7 — Персональна сторінка користувача з спробами та часом доступу до приміщення

Сторінка з викладачами має аналогічний функціонал. Сторінка з корпусами має наступний вигляд та вже знайомий функціонал з додавання нових сутностей.

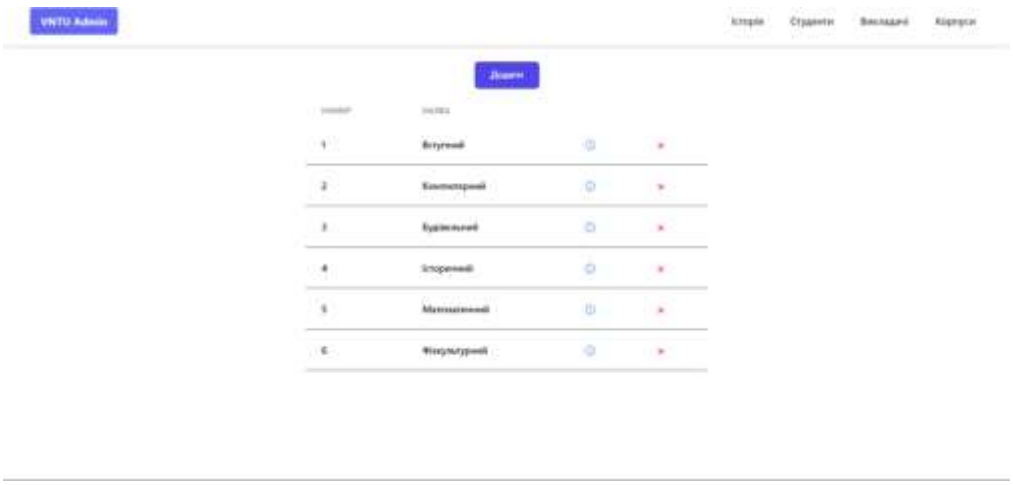
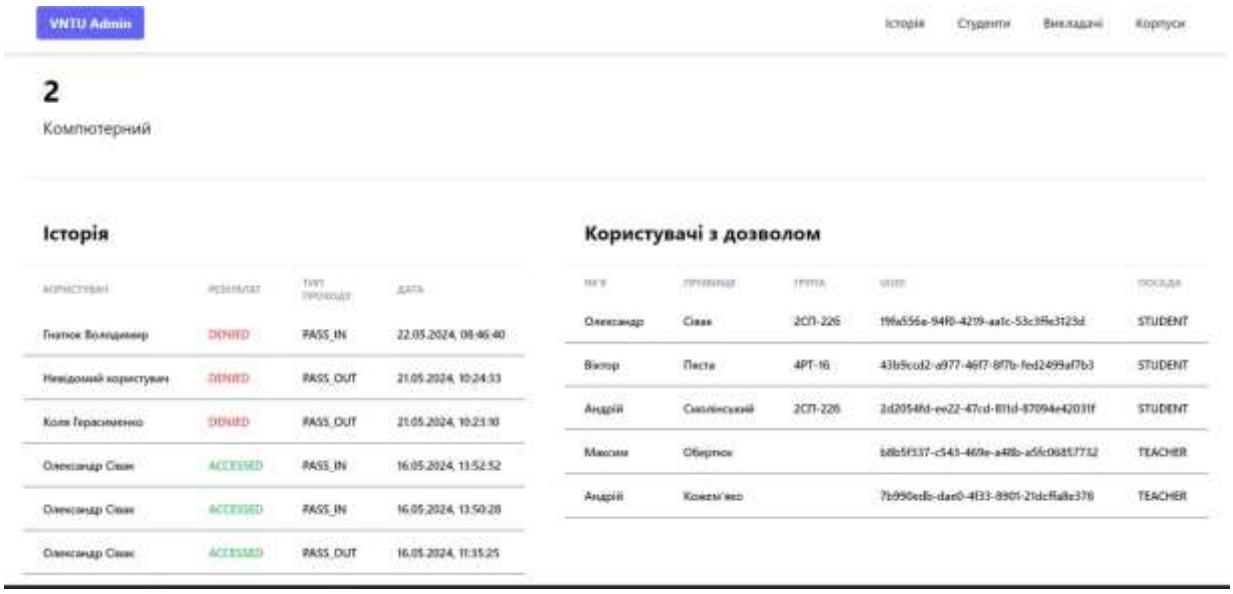


Рисунок 3.8 — Сторінка з інформацією про приміщення з можливістю редагування

Персональна сторінка корпусу з відповідною інформацією:



На даному рисунку представлено історію доступу користувачів. Користувачами є студенти або співробітники, які мають відповідний рівень доступу до визначеного корпусу університету. Крім того показуються результати спроби отримати доступ до приміщення, а саме дозволено чи заборонено визначено чи було проходження в корпус. Також визначається час доступу та дата. Програмно реалізована функція визначення, які саме користувачі володіють доступом до приміщення. Вони визначаються із бази даних, а саме такі атрибути: ім'я, прізвище та група. Кожному з користувачів. надається унікальний код UUID які є набором шестицяткових чисел і є унікальним та неповторним.

4 ДОСЛІДЖЕННЯ Й ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Принципи тестування

Розроблена система в частині серверної бази даних не має модульних тестів, хоча існує простий інтеграційний тест, а також безперервна перевірка адекватності ECHOTEST, включена в неї. Юніт-тести і більш комплексні інтеграційні тести полегшили б розробку. Планується досягти 100% покриття юніт- тестами і налаштувати безперервну інтеграцію. Якщо необхідно, щоб система працювала, необхідно постійно контролювати її. Зокрема, визначати метрики, що оцінюють стан і продуктивність системи, повинні експортуватися; при виникненні проблеми повинні автоматично виконуватися дії, які можна вжити автоматично; а дії, що вимагають втручання людини, повинні сповіщати людину про це.

Спосіб моніторингу системи та вжиття заходів. Необхідно знайти відповідні дії (в ідеалі - на основі існуючого загального рішення). Деякі базові функції сторожового таймера присутні у самому Deadlock: контролери мають апаратний сторожовий таймер, який перезапускає їх у разі блокування, а тест інтеграції, включений у deadaux, може попередити людину, якщо щось явно не працює. Однак ми маємо намір дослідити більш комплексні рішення. Високорівневі правила та інтерфейс оптимізовані для використання в університетах. В рамках розгортання в нашому університеті буде розроблено модель правил для конкретного домену та створено відповідний інтерфейс управління правилами.

Інструменти та бібліотеки. Повинні бути надані інструменти та бібліотеки, які ще більше полегшить розгортання системи та її інтеграцію. Зокрема, інструмент для імпорту даних з часто використовуваних систем, таких як бази даних каталогів з використанням протоколу LDAP або бази даних SQL, буде доступний до розгортання в масштабах всієї організації, яке заплановане найближчим часом.

4.2 Налаштування клієнтського застосунку LOKKYY APP

Завантажити додаток LOKKYY APP з сервісу Google Play Store або Apple App Store та авторизуватись у ньому.

У смартфоні увімкнути бездротовий інтерфейс Bluetooth розташувати його на відстані на більше ніж 3 м відпристрою LOKKYY. Не потрібно закривати програму та не відходити далі від пристрою LOKKYY до завершення процедури налаштування.

При налаштуванні комплекса LOKKYY зі смартфона з операційною системою IOS можливе наступне використання інтерфейсу Bluetooth, як з IOS так і з Android смартфонів. При налаштуванні з Android смартфонів, використання Bluetooth буде можливим лише на Android смартфонах.

Запустити додаток LOKKYY APP. На головному екрані обрати: «Налаштувати новий пристрій» (рис. 4.1 а). Потрібно обрати пристрій зі списку (рис. 4.1 б), внести дані до профілю та назвати пристрій (рис. 4.1 в). У разі налаштування пристрою LOKKYY для двох приводів потрібно активізувати перемикач «Другі ворота» (рис. 4.1 г).

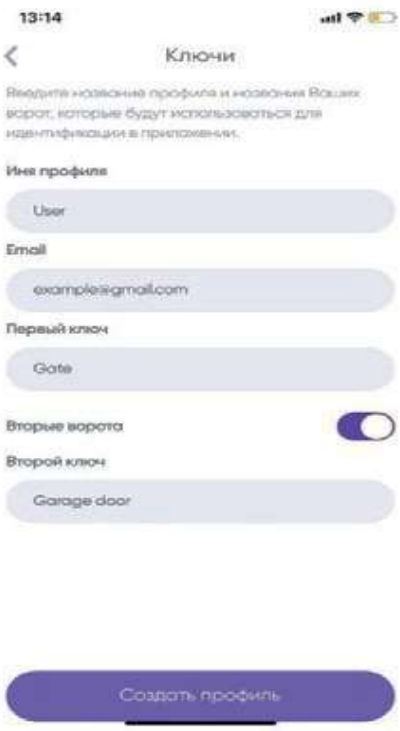
Далі слід натиснути кнопку «Створіть профіль» та підключити пристрій до мережі Wi-Fi для доступу в Інтернет (якщо така мережа доступна в зоні воріт) або пропустити цей крок, натиснувши кнопку «Skip» (рис. 4.2).

Рекомендується одразу підключити мережу Wi-Fi та доступ в Інтернет, щоб користуватись усіми функціональними можливостями. Якщо пристрій LOKKYY підключити лише по Bluetooth, у користувача буде відсутній повний функціонал додатка – користуватись можна буде лише одним ключем, оскільки нові ключі створюються лише по Wi-Fi. Також будуть відсутні логи історії та пуш –повідомлення.

а)

б)

в)



г)

Рисунок 4.1 — Налаштування пристрою LOKKYY в додатку LOKKYY APP

Рисунок 4.2 — Підключення пристрою LOKKYY до мережі Wi-Fi

4.3 Тестування роботи комплекса LOKKYY

Перевірка роботи комплекса LOKKYY в режимі керування

Після успішного налаштування додатка LOKKYY APP у центрі головного екрана буде доступна велика кнопка відкривання, а знизу під нею - закривання кнопку як на рис. 4.3.

Рисунок 4.3 — Вигляд головного екрану додатка LOKKYY APP в робочому режимі керування

Після того, як переконаєтеся, що на шляху відчинення воріт немає перешкод, натисніть на велику кнопку і огляньте, чи спрацював привід воріт (ролет, шлагбаума тощо). При цьому стан відкриття / закриття воріт повинен змінитися на «ВІДКРИТО». Якщо пристрій не працює, уважно перевірте правильність монтажу та налаштування.

4.3.1 Користувацькі налаштування додатка LOKKYY APP

Для входу в меню налаштувань на головному екрані (рис. 11) слід натиснути кнопку . Всі опції меню налаштувань інтуїтивні зрозумілі та зручні в користуванні (рис. 4.4).

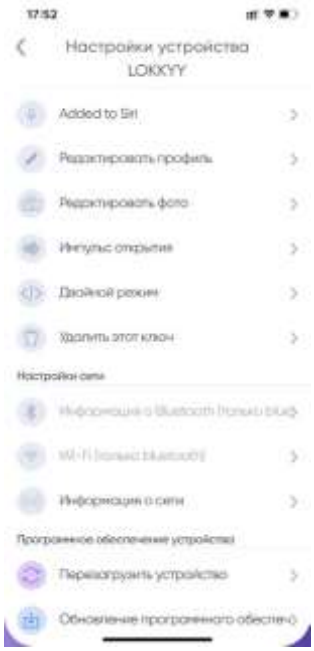
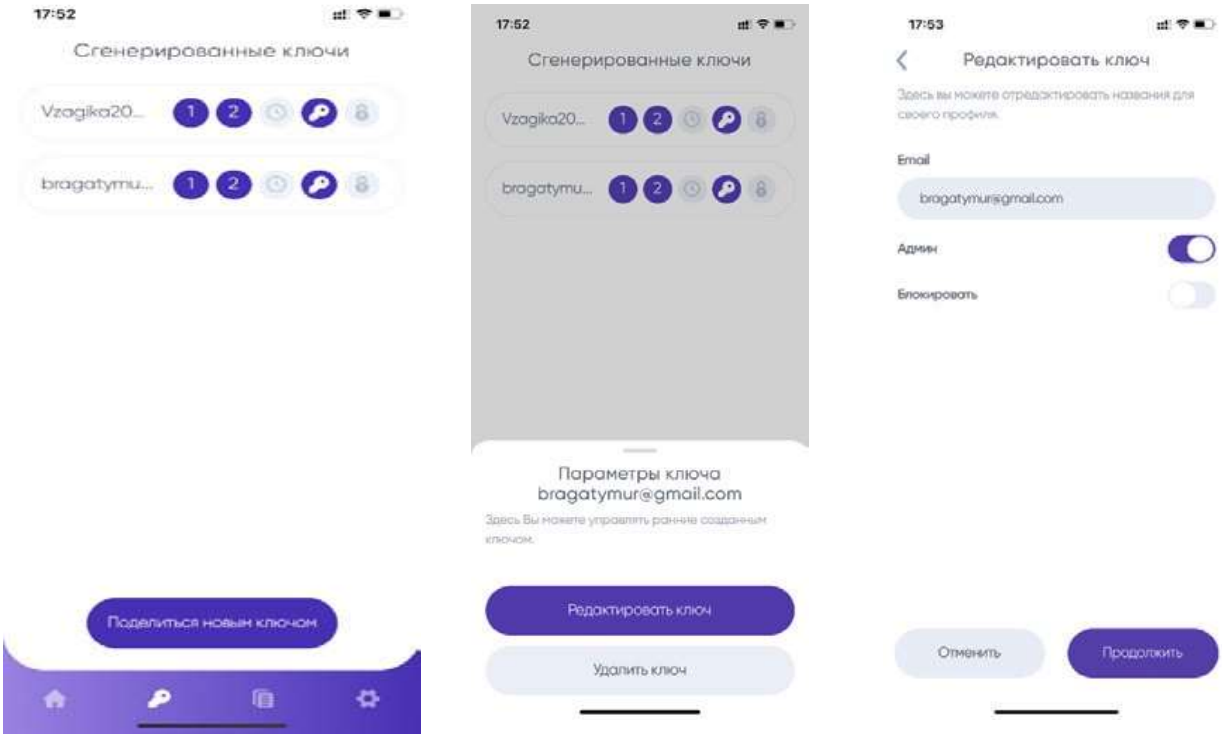


Рисунок 4.4 — Меню налаштувань додатка LOKKYY APP

4.3.2 Генерування й поширення ключа доступу

Натисніть кнопку  в нижній частині додатка для входу в меню управління ключами доступу. В цій закладці відображаються наявні ключі доступу (рис. 4.5 а). При натисканні на вже згенерований ключ з’являється можливість редагування цього ключа (рис. 4.5 б) або його видалення (рис. 4.5 в).



а) б) в)

Рисунок 4.5 Редагування ключів доступу

Для генерування нових ключів доступу натисніть «Поділитися ключем» (рис.4.6).

Рисунок 4.6 — Генерування й поширення нового ключа

4.3.3 Налаштування теми додатка

Після натискання кнопки в нижній частині додатка доступний вибір кольорової теми інтерфейсу користувача (рис. 4.7).

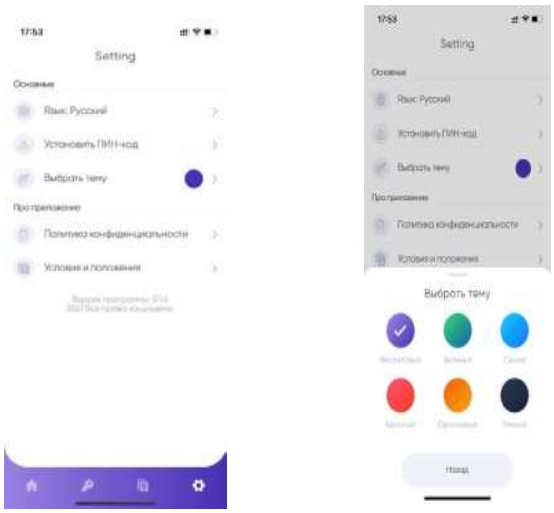


Рисунок 4.7 — Вибір кольорової теми додатка

4.4 Комплексна перевірка функціонування комплекса LOKKYY

4.4.1 Перевірка роботи комплекса LOKKYY в режимі керування

Після успішного налаштування додатка LOKKYY APP у центрі головного екрана буде доступна велика кнопка відкривання, а знизу під нею - закривання кнопку як на рис. 4.8.

Рисунок 4.8 — Вигляд головного екрану додатка LOKKYY APP в робочому режимі керування

Після того, як переконаєтеся, що на шляху відчинення воріт немає перешкод, натисніть на велику кнопку і огляньте, чи спрацював привід воріт (ролет, шлагбаума тощо). При цьому стан відкриття / закриття воріт повинен змінитися на «ВІДКРИТО». Якщо пристрій не працює, уважно перевірте правильність монтажу та налаштування.

4.4.2 Користувацькі налаштування додатка LOKKYY APP

Для входу в меню налаштувань на головному екрані (рис. 11) слід натиснути кнопку . Всі опції меню налаштувань інтуїтивні зрозумілі та зручні в користуванні (рис. 4.9).

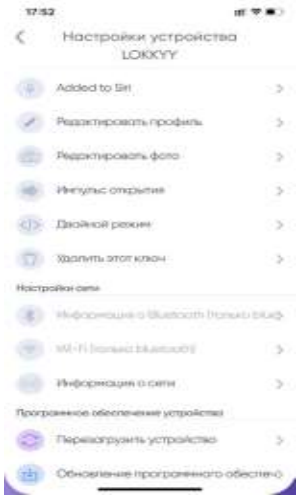
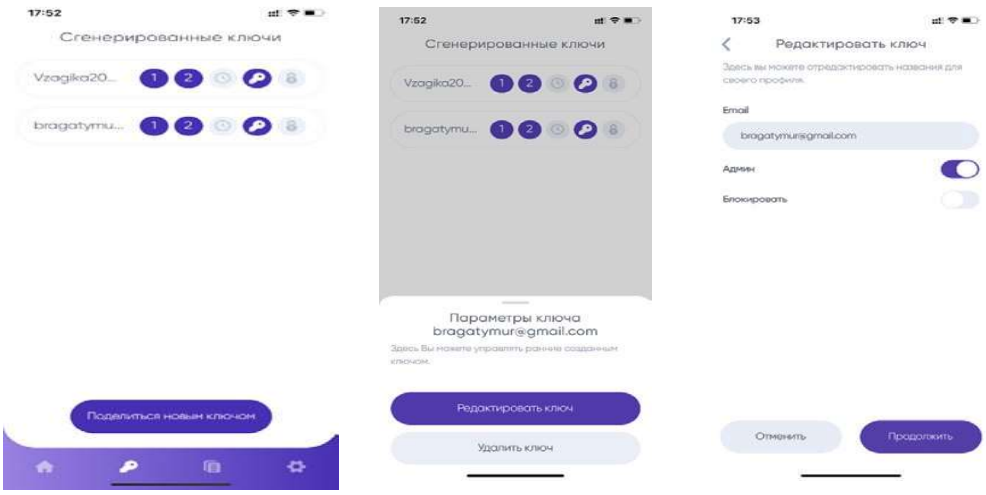


Рисунок 4.9 – Меню налаштувань додатка LOKKYY APP

4.4.3 Генерування й поширення ключа доступу

Натисніть кнопку в нижній частині додатка для входу в меню управління ключами доступу. В цій закладці відображаються наявні ключі доступу (рис. 4.10 а). При натисканні на вже згенерований ключ з’являється можливість редагування цього ключа (рис. 4.10 б) або його видалення (рис. 4.10 в).



а) б) в)

Рисунок 4.10 — Редагування ключів доступу

Для генерування нових ключів доступу натисніть «Поділитися ключем» (рис. 4.11).

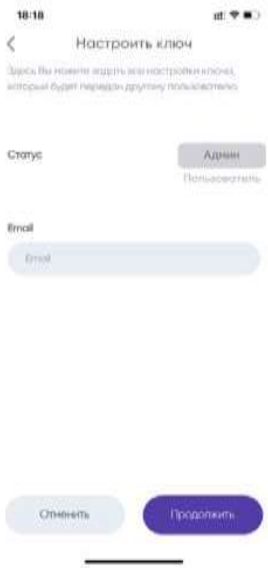


Рисунок 4.11 – Генерування й поширення нового ключа

Після натискання кнопки  в нижній частині додатка доступний вибір кольорової теми інтерфейсу користувача (рис. 4.12).

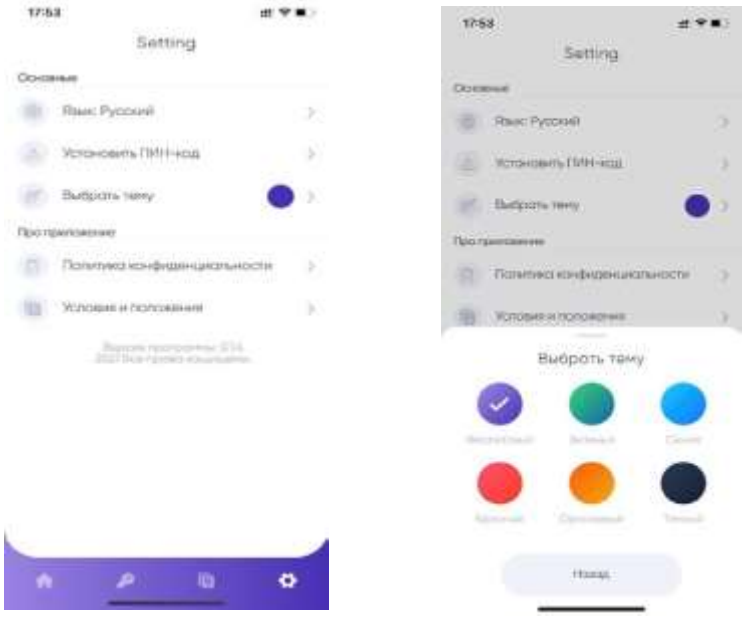


Рисунок 4.12 – Вибір кольрової теми додатка

4.5 Перевірка функціональних вимог до програмного забезпечення

Визначення основних функціональних можливостей (додаток А) комплекса LOKKYY проводять під час налаштувань і використання додатка LOKKYY APP в попередніх пунктах випробувань апаратної частини цієї комплексної магістерської роботи. (кроки 1-6 настанов з експлуатації).

Перевірку вимог до вмонтованого ПЗ комплекса LOKKY та його збереження проводять шляхом експериментального дослідження таких можливостей:

1) встановлення й робота на всіх смартфонах з операційними системами не нижче Android 4.0 або IOS 11;

2) робота в складі комплекса LOKKYY з доступом до Інтернет-мережі, а при її відсутності - за допомогою Bluetooth;

3) авторизація й створення профіля користувача з його логіном, паролем, фото;

4) доступ до застосунку з використанням варіантів розблокування, заданого користувачем;

5) налаштування пристрою LOKKYY з меню застосунка через Bluetooth, створення профілю та під'єднання пристрою до Wi-Fi;

6) перевірка роботи комплекса LOKKYY при налаштуванні;

7) керування ключами доступу – генерування, редагування, передавання іншим користувачам, видалення;

8) перегляд поточного стану об'єктів і інформації про час і користувачів, що керували пропуском;

9) налаштування кольорової теми додатка;

10) оновлення додатка LOKKYY APP із сервісів Google Play Store та Apple App Store;

11) захист ПЗ від несанкціонованих змін в процесі експлуатації.

Вказані експериментальні дослідження проводять шляхом завантаження, налаштування й пробного використання мобільного додатка LOKKYY мінімум на двох смартфонах з ОС IOS та Android з використанням меню додатка.

ПЗ комплекса LOKKYY вважають таким, що відповідає вимогам , якщо в процесі експериментальних досліджень підтверджено всі функціональні можливості, а ПЗ функціонує без помилок і збоїв.

Перевірка технічних параметрів і протоколів бездротового передавання даних в стандартах Bluetooth та Wi-Fi на їх заводах-виробниках вмонтованих пристроїв та підтверджується гарантією на відповідні комплектуючі. Перевірка бездротового передавання даних в цих ТУ полягає в комплексній перевірці апаратної платформи пристрою LOKKYY та програмного забезпечення LOKKYY APP, що підтримують взаємодію по мережі Bluetooth та доступ до мережі Інтернет за допомогою з'єднань Wi-Fi.

Комплекс LOKKYY вважають таким, що витримав випробування на відповідність вимогам до бездротових інтерфейсів Bluetooth та Wi-Fi, якщо в процесі перевірок не зафіксовано збоїв, переривань чи інших несправностей зв'язку в зоні їх дії.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового мобільного застосуноку для смартфона (клієнтська частина) та бази даних (серверна частина) для комплексу мобільного керування автоматикою пропуску на об’єкти. Особливістю розробки є універсальність і мобільність завдяки функцірнальним можливостям контролю й керуванням через мережі Інтернет, Bluethooth, Wi-Fi за допомогою відповідного мобільного додатка. Цим забезпечується відсутність будь-яких апаратних ключів, відсутність додаткового апаратного сервера (пристрою централізованого керування), простота надання довготривалого й короткотермінованого доступу, оперативність зміни ключів, розширена функціональність віддаленого керування й моніторингу. Аналогом може бути система контролю та управління доступом Evertech – 100 000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-те-	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Продовження табл. 5.1

Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

Продовження табл. 5.1

10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	3	4
Цінова політика	3	4	3
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	3	3
Супровідна документація	3	3	4
Сума	41	40	42
Середньоарифметична сума балів	(41+40+42) / 3 = 41		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів,	Рівень комерційного потенціалу
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що універсальність і мобільність завдяки функцірнальним можливостям контролю й керуванням через мережі Інтернет, Bluethooth, Wi-Fi за допомогою відповідного мобільного додатка. Цим забезпечується відсутність будь-яких апаратних ключів, відсутність додаткового апаратного сервера (пристрою централізованого керування), простота надання довготривалого й короткотермінованого доступу, оперативність зміни ключів, розширена функціональність віддаленого керування й моніторингу.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$З_о = \frac{M}{T_р} \cdot t, \quad (5.1)$$

де М – місячний посадовий оклад конкретного розробника (дослідника), грн.;

$Т_р$ – число робочих днів за місяць, 20 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.4.

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	25000	1250,00	28	35000,000
Програміст	24000	1200,00	28	33600,000
Всього				68600,00

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 10 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 10\ \% / 100\ \% \qquad (5.2)$$

$З_д = (68600{,}00 \cdot 10\ \% / 100\ \%) = 6860{,}00$ (грн.)

5.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22\ \% / 100\% \qquad (5.3)$$

$Н_з = (68600{,}00 + 6860{,}00) \cdot 22\ \% / 100\ \% = 16601{,}20$ (грн.)

5.2.4. Оскільки для розроблювального пристрою не потрібно витрачати матеріали та комплектуючи, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{Тв} \cdot \frac{t_{вик}}{12} \text{ [грн.].} \quad (5.4)$$

де Ц – балансова вартість обладнання, грн.;

Т – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 35000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,40 міс.

$$A_{обл} = \frac{35000}{2} \times \frac{1{,}4}{12} = 2041{,}67 \ грн.$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.5. Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $В_{нем.ак.}$ = 15000 грн.

Таблиця 5.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортиза-ційні відрахування-ня, грн.
Комп'ютер та комп'ютерна периферія	35000	2	1,40	2041,667
Офісне обладнання (меблі)	20000	4	1,40	583,333
Приміщення	800000	20	1,40	4666,667
Всього				7291,67

5.2.6 Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \quad (5.5)$$

де В – вартість 1 кВт-години електроенергії для 1 класу підприємства з ПДВ в 2024 році для Вінницької області за даними Енера-Вінниця, В = (5635,47/1000)*1,2 = 6,76 грн./кВт;

П – встановлена потужність обладнання, кВт. П = 0,28 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

$К_п$ – коефіцієнт використання потужності, $К_п$ = 0,9.

$В_е$ = 0,9 · 0,28 · 8 · 28 · 6,76 = 381,58848 (грн.)

5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (5.6)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

$I_в$ = 68600,00 * 55% / 100% = 37730 (грн.)

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників:

$$Н_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (5.7)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$H_{нзв}$ = 68600,00 * 115 % / 100 % = 78890 (грн.)

5.2.9 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$B_{заг}$ = 68600,00+6860,00+16601,20+7291,67+15000+381,59+37730+
+78890 = 231354,46 грн.

5.2.11 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \qquad (5.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η=0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7; впровадження, то η=0,9. Оберемо η = 0,5, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$ЗВ$ = 231354,46 / 0,5 = 462709 грн.

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

— абсолютного економічного ефекту (чистого дисконтованого доходу);

— внутрішньої економічної дохідності (внутрішньої норми дохідності);

— терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta П_i = (\pm \Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \qquad (5.9)$$

де $\pm\Delta Ц_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_o$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $Ц_o = Ц_б \pm \Delta Ц_o$;

$Цб$ – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0{,}8333$.

ρ – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2024 році ϑ =18%.

Припустимо, що при прогнозованій ціні 25000 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на *2000* грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 1000 шт., протягом другого року – на 1300 шт., протягом третього року на 1500 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

*$\Delta П_1$ = (0*2000 + (*25000 + *2000)*1000)* 0,8333* 0,21) * (1 - 0,18) = 3587499,857 грн.*

*$\Delta П_2$ = (0*2000 + (*25000 + *2000)*(1000+1300)* 0,8333* 0,21) * (1 - 0,18) = 8911349,644 грн.*

*$\Delta П_3$ = (0*2000 + (*25000 + *2000)*(1000+1300+1500)* 0,8333* 0,21) * (1 - 0,18) = 14723099,411 грн.*

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 27221948,91 грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (5.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

Т – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05...0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$ПП = (3587499,857/(1+0,1)^1)+(8911349,644/(1+0,1)^2)+(14723099,411/(1+0,1)^3) = 3261363,51 + 7364751,772 + 11061682,5 = 21687797,78 \text{ грн.}$$

Далі розраховують величину початкових інвестицій *PV*, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{інв} * ЗВ, \quad (5.11)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{інв}=2...5$, але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 462709 = 925417,82 \text{ грн.}$$

Тоді абсолютний економічний ефект *Еабс* або чистий приведений дохід *(NPV, Net Present Value)* для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП – PV, \quad (5.12)$$

$E_{абс}$ = 21687797,78 – 925417,82 = 20762379,96 грн.

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої но-рми дохідності *(IRR, Internal Rate of Return)* вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього використаємо формулу:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{5.13}$$

$T_ж$ – життєвий цикл наукової розробки, роки.

$E_в = \sqrt[3]{(1 + 20762379,96/925417,82} - 1 = 1,862$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \tag{5.14}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні d = (0,09…0,14);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,5).

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_в > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в}, \qquad (5.15)$$

$T_{ок}$ = 1 / 1,862 = 0,54 р.

Оскільки $T_{ок}$ < 3-х років, а саме термін окупності рівний 0,54 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 462709 гривень. Було зпрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкуренто спроможним. Період окупності складе близько 0,54 роки.

ВИСНОВКИ

В комплексній магістерській кваліфікаційній роботі, її програмній частині виконано такі завдання:

— огляд і аналіз програмно-апаратних засобів керування доступом;

— аналіз та вибір технологій проектування програмного забезпечення комплексу керування пропуском;

— розробка бази даних керування пропуском для серверного програмного забезпечення

— адаптація застосунку користувача до розробленої бази даних;

— комплексне дослідження й тестування розробленого програмного продукту;

— розрахунок економічних показників проєкту.

Створене серверне програмне забезпечення спеціалізованої бази даних дозволило побудувати гнучку ієрархічну систему віддаленого керування доступом через веб-інтерфейс для будь-якого об'єкта на основі розробленого апаратного пристрою керування.

Отже, всі поставлені в роботі завдання успішно виконано, а мету роботи, яка полягає в розширенні функціональних можливостей комплексу мобільного керування автоматикою пропуску з використанням безпровідних технологій, інтегрованих з програмним забезпеченням смартфонів та інтернет-мережі – досягнуто.

За матеріалами роботи опубліковано тези доповіді [1].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. АПАРАТНО-ПРОГРАМНИЙ КОМПЛЕКС МОБІЛЬНОГО ДИСТАНЦІЙНОГО КЕРУВАННЯ ДОСТУПОМ / Гуменюк В.В., Зубринська Д. Л., Крупельницький Л.В., Городецька О.С // Міжнародна науково-практична Інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи (МН2024)» (15 жовтня 2023 р.- 20 травня 2024 р., Вінниця) : Режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/21172/17557

2. Системи контролю та управління доступом. [Електронний ресурс]. Режим доступу: https://evertech.ua/access-controll-systems/.

3. Технічні системи безпеки на об’єктах. [Електронний ресурс]. Режим доступу: https://karabiner.ua/ua/stati/tehnicheskie-sistemy-bezopasnosti-na-obektah/

4. Системи контролю і управління доступом від А до Я. [Електронний ресурс]. Режим доступу: https://deps.ua/ua/knowegable-base/referenceinformation/7824.html.

5. Системи контролю та управління доступом. [Електронний ресурс]. Режим доступуhttps://www.rim2000.com/equipment/networks/access-control/.

6. СКУД - система контролю та управління доступом. [Електронний ресурс]. Режим доступу: http://www.centrebespek.com/articles/ELEMENT_ID_14787/.

7. Gean Davis Breda New Era of Mobile Access Control System / Gean Davis Breda, Raul Mariano Cardoso, Felipe André Cordeiro Pirotta // International Journal of Computer Science and Network Security, VOL.15, No.8, 2015, P. 6 – 15.

8. Rajashree S. Bluetooth and NFC Enabled Contactless Access Control System / S.Rajashree, S. Kaushik, K. Varman // ScieXplore: International Journal of Research in Science, 2015, № 2, P. 1 □ 32.

NFC: умные метки. 9. [Електронний ресурс]. Режим доступу: https://itc.ua/articles/nfc_umnye_metki_53544/.

10. Бідюк П. Сучасні методи біометричної ідентифікації / Петро Бідюк, Володимир Бондарчук // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. 2009. Випуск 1(18). С. 137 □ 146.

11. Царьов Р.Ю. Біометричні технології: навч. посіб. [для вищих навчальних закладів] / Р.Ю. Царьов, Т. М. Лемеха. □ Одеса: ОНАЗ ім. О.С.
Попова, 2016. □ 140 с.

12. Контроллер доступа NDC F18IP(U-Prox IP400). [Електронний ресурс].
Режим доступу: https://www.forter.com.ua/kontrollery-dostupa/u-prox-ip-400/.

13. Зчитувач-контролер SameKey Card Control. [Електронний ресурс].
Режим доступу: https://secur.ua/kontroller-schityvatel-samekey-card-control.html.

14. Комплекс мобільного керування автоматикою пропуску LOKKYY[Електронний ресурс]. Режим доступу: https://lokkyy.com/#about.

15. Створення безпечної аутентифікації користувачів у веб-додатках. [Електронний ресурс]. Режим доступу: https://redstone.media/stvorennya-bezpechnoyi-autentifikaciyi-koristuvachiv-u-veb-dodatkah

16. Ідентифікація, автентифікація та авторизація — як не передати керування доступами зловмисникам. [Електронний ресурс]. Режим доступу: https://thekernel.ua/identyfikatsiia-avtentyfikatsiia-ta-avtoryzatsiia/

17. Організація бездротових мереж. [Електронний ресурс]. Режим доступу: https://sez.net.ua/organizaciya-komp-yuternyh-setej-38-2.html

18. Комп’ютерні мережі : підручник / Азаров О. Д., Захарченко С. М., Кадук О. В. та ін.. − Вінниця : ВНТУ, 2020, 378 с.

ДОДАТОК А Технічне завдання

Міністерство освіти та науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

проф., д.т.н.. Азаров О.Д. _____

" ____ " _____ 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання комплексної магістерської кваліфікаційної роботи

"Комплекс мобільного керування автоматикою пропуску".

Частина 2. "Програмна частина"

08-54.КМКР.008.00.000 ТЗ

Науковий керівник: доцент к.т.н.

_____ Городецька О.С.

Виконав: студентка групи КІ-22мз

_____ Зубринська Д.Л.

ВНТУ 2024

1 Підстава для виконання магістерської кваліфікаційної роботи (МКР) - наказ по ВНТУ № від

1.1 Необхідність побудови гнучкої та легко впроваджувальної системи віддаленого керування правами доступу з підтримкою можливості надання доступу в автономному режимі з використанням технологій мобільної ідентифікації.

1.2 Наказ про затвердження теми МКР.

2 Мета МКР і призначення розробки

2.1 Мета робота — розширення функціональних можливостей системи віддаленого керування доступом через веб-інтерфейс за рахунок підтримки можливості роботи в офлайн режимі;

2.1 Призначення розробки — визначення підходів до побудови апаратних та програмних засобів системи віддаленого керування доступом через вебінтерфейс.

3 Вихідні дані для виконання МКР

3.1 Функціональне призначення — віддалене керування доступом з використанням хмарних та мобільних технологій;

3.2 Організація доступу - за ідентифікатором, що надсилається зі смартфону або планшету;

3.3 Реєстрація в системі — відділена через веб-інтерфейс;

3.4 Інтерфейси - Ethernet, WI-FI та Bluetooth;

3.5 Протоколи - TCP/IP, HTTP, Bluetooth;

3.6 Вихід контролера — 2 комутовані канали потужністю до 4 КВт;

3.7 Вхід контролера — 2 оптично розв′язані канали;

3.8 Живлення контролера — джерело постійної напруги +5 В потужністю до 3 Вт.

4 Вимоги до виконання МКР

4.1 Провести обґрунтування доцільності розробки;

4.2 Провести аналіз сучасних технологій управління доступом;

4.3 Визначити підходи до реалізації апаратних та програмних засобів системи віддаленого керування доступом через веб-інтерфейс з підтримкою можливості роботи в режимі офлайн;

4.4 Оцінити комерційний потенціал розробки.

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасних сучасних технологій управління доступом	15.03.2024	20.03.2024	Вступ Розділ 1
2	Огляд принципів та технологій віддаленого керування доступом	21.03.2024	28.03.2024	Розділ 2

3	Розробка структурної схеми програмного забезпечення	01.04.2024	18.04.2024	Розділ 3.1,
4	Аналіз технологій можливої реалізації	19.04.2024	27.04.2024	Розділ 3.2
5	Розробка і тестування алгоритмів роботи системи та її компонентів. Розробка вебсерверу, додатку адміністратора та мобільного додатку користувача	18.05.2024	27.05.2024	Розділ 3,4 Блок схеми алгоритмів, додатки
6	Оцінка комерційного потенціалу розробки	28.05.2024	30.05.2024	Розділ 5
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	01.06.2024	04.06.2024	Пояснювальна записка, графічний матеріал, презентація

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам. Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

7 Вимоги до оформлювання та порядок виконання МКР

7.1 При оформлюванні МКР використовуються:

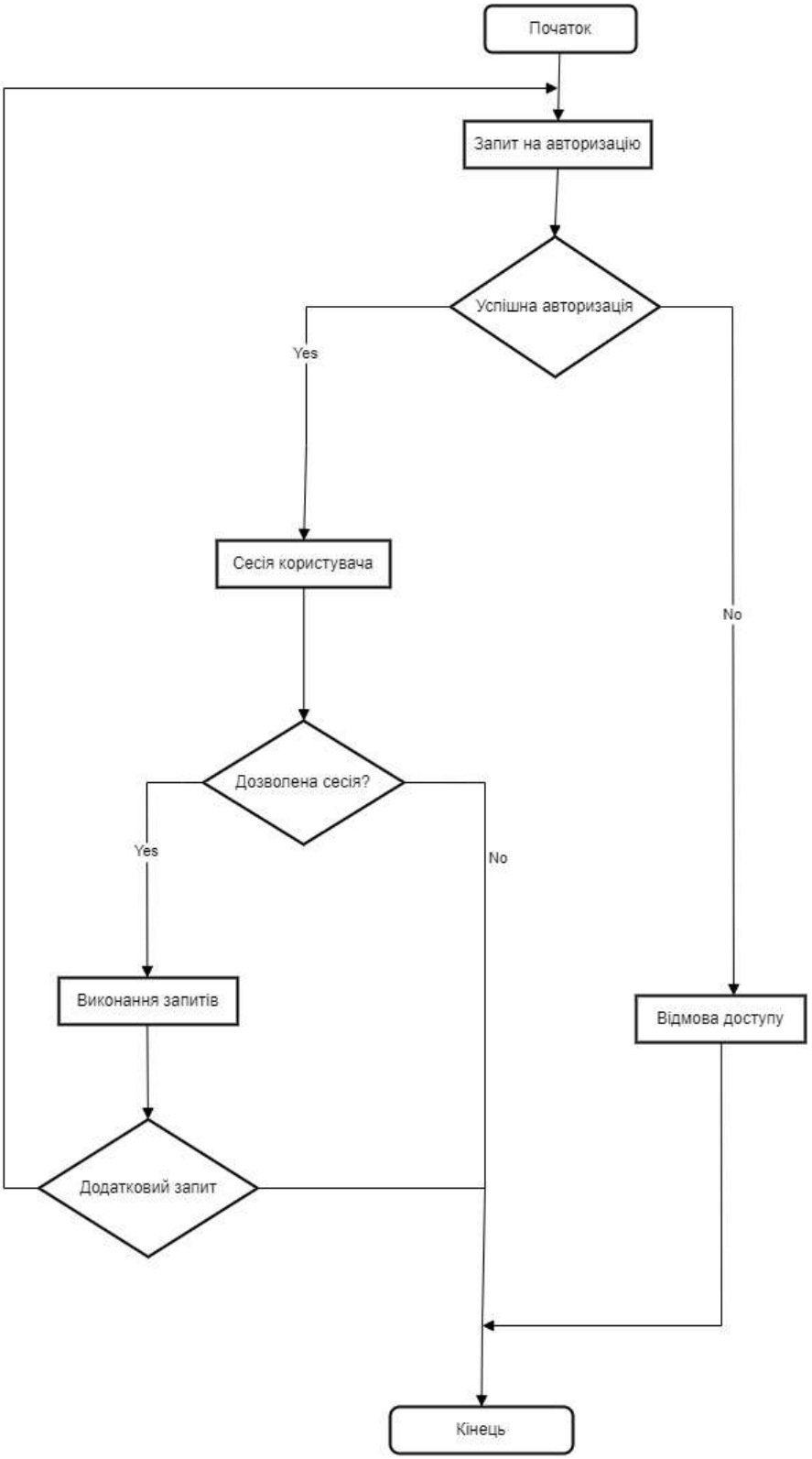
— ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення т правила складання»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт студентами спеціальності 123 «Комп'ютерна інженерія». / Укладачі О.Д. Азаров, О.В. Дудник, С.І. Швець − Вінниця : ВНТУ, 2023. − 57 с.

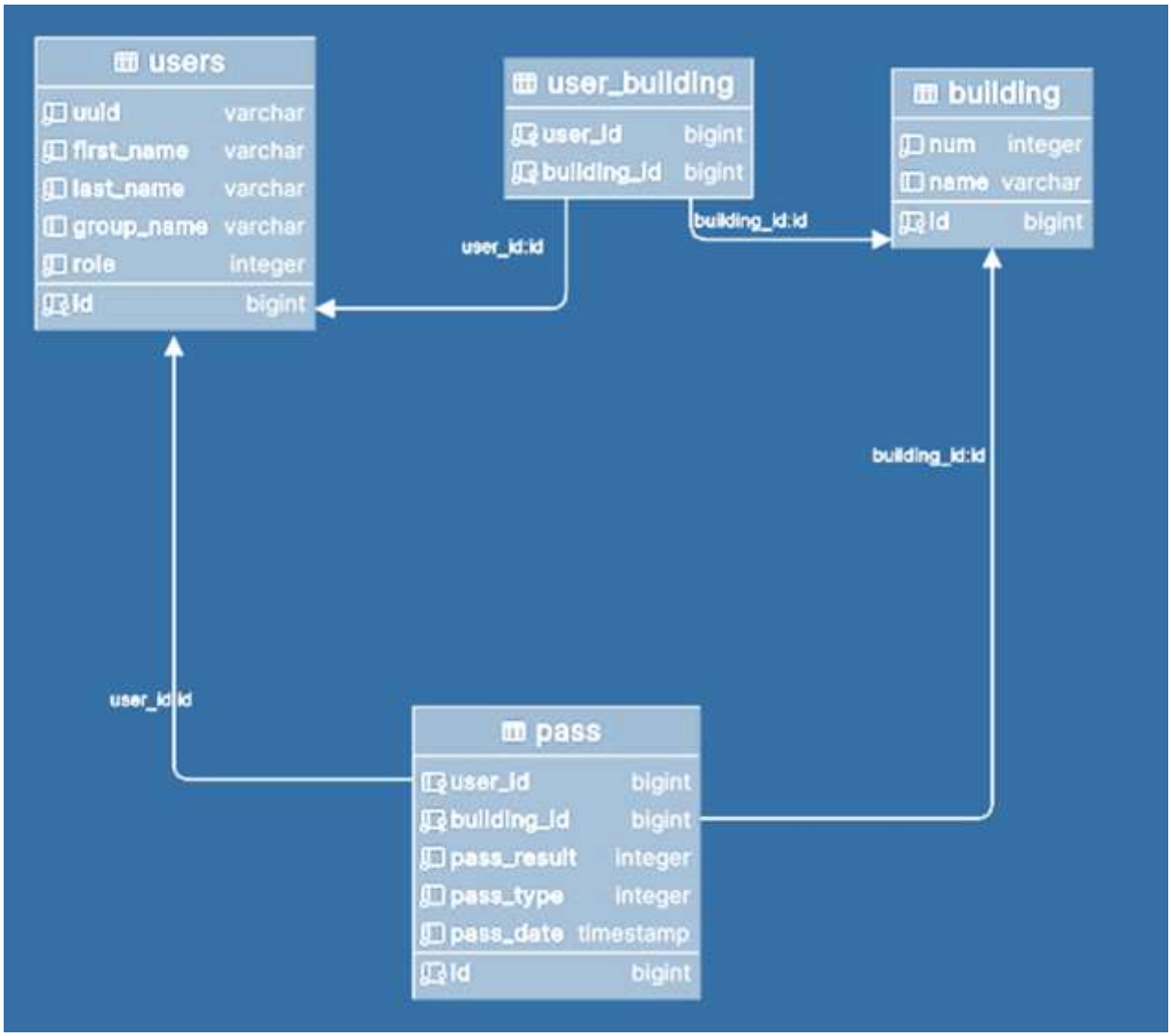
7.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

Додаток Б. Блок-схема авторизації

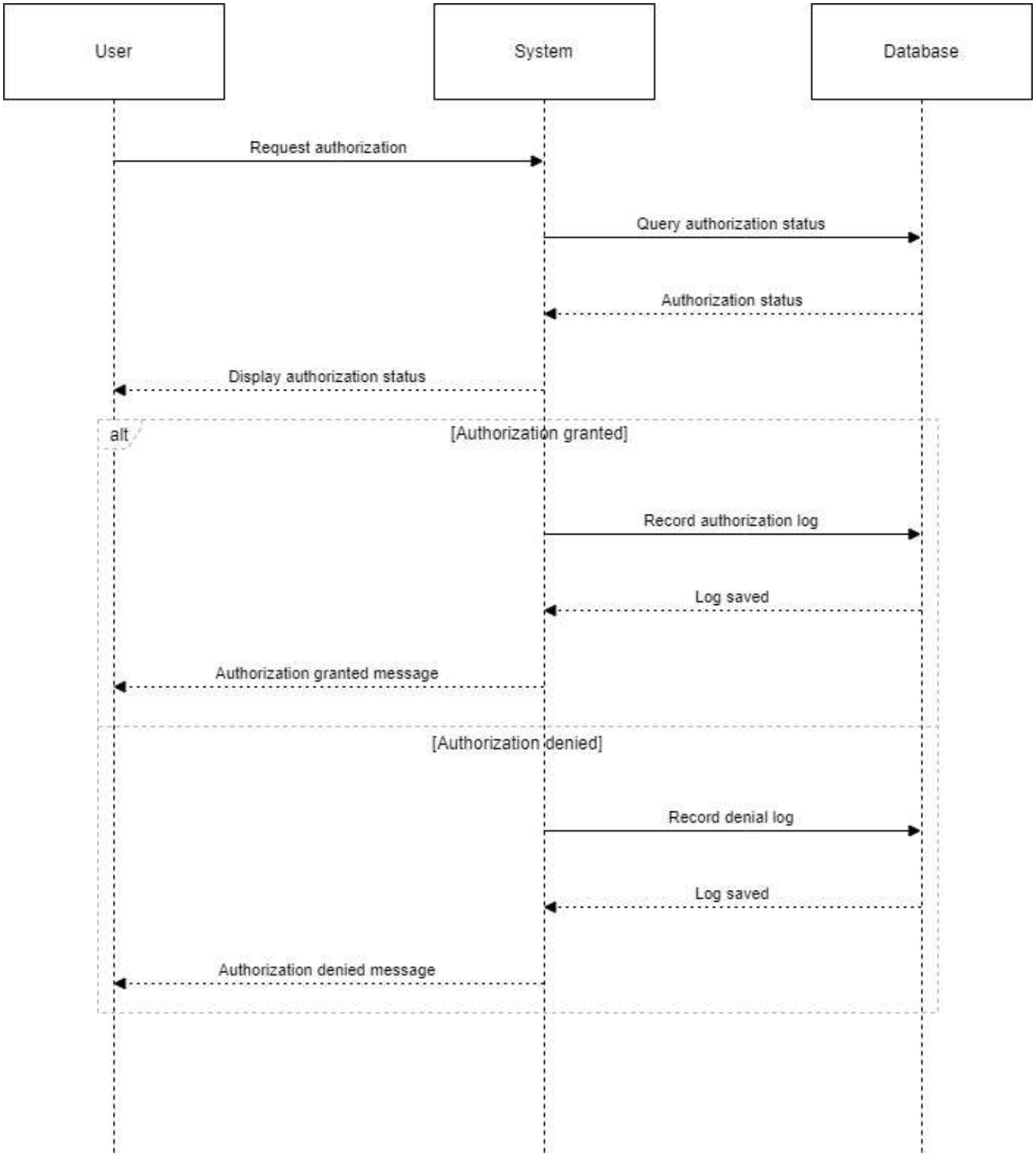


Yes

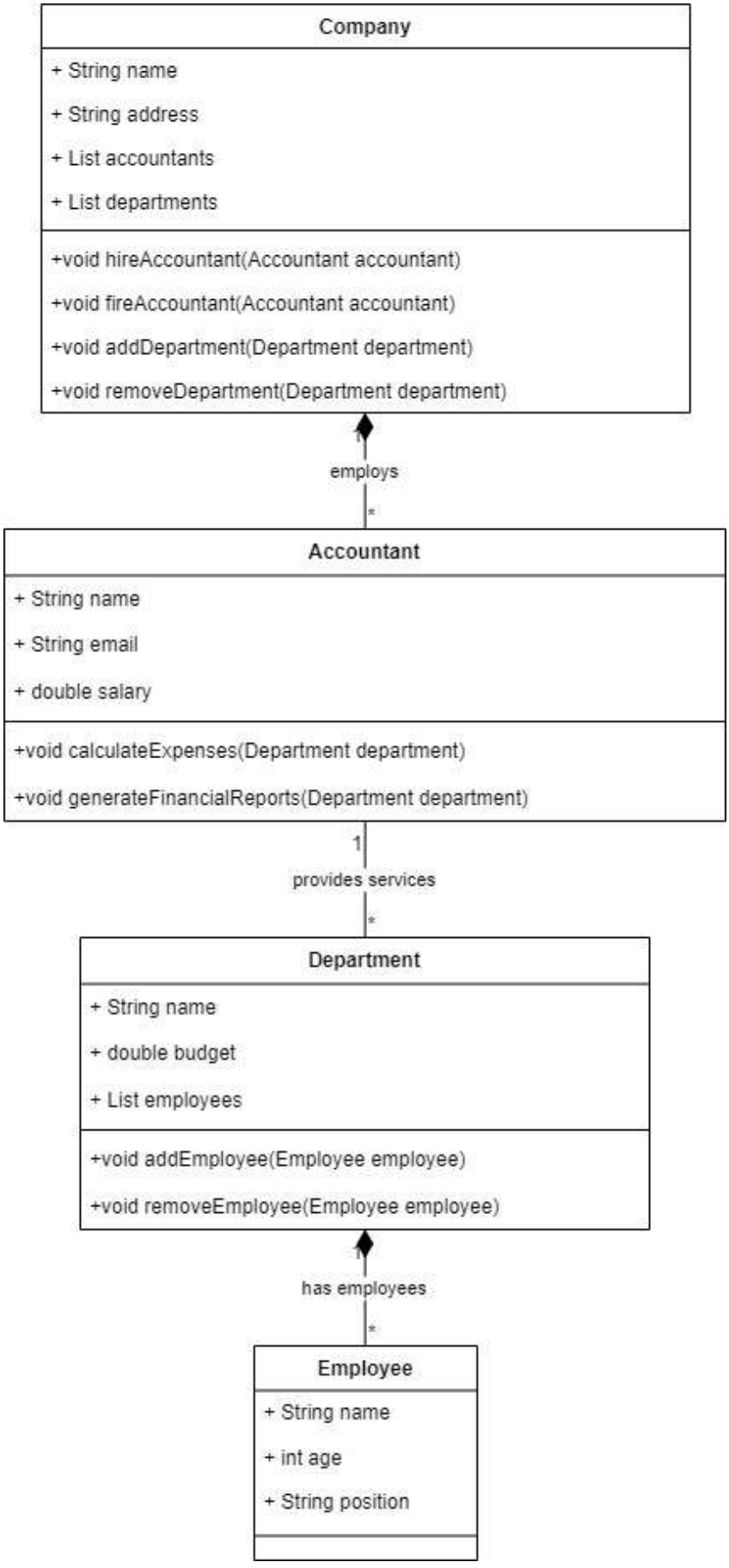
Додаток В. ER-діаграма «сутність-зв’язок» розробленої бази даних



Додаток Г. Процес авторизації на веб-сервісі користувача



Додаток Д. UML діаграма веб сервісу керування допуском



Додаток Е. Код програми

App.jsx 1 | index.js | Article.jsx | Aside.jsx | Header.jsx

src > index.js > ...

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import { HashRouter } from 'react-router-dom';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <HashRouter>
      <App />
    </HashRouter>
  </React.StrictMode>
);

reportWebVitals();
```

src > App.jsx > ...

```
import { Routes, Route } from 'react-router-dom';
import Aside from './components/Aside/Aside';
import Header from './components/Header/Header';
import Article from './components/Main/Article';

function App() {

  return (
    <div className="App">
      <Header></Header>
      <Routes>
          <Route path='/aside' element={<Aside/>}/>
          <Route path='/article' element={<Article/>}/>
      </Routes>
    </div>
  );
}

export default App;
```

components > Header > Header.jsx > default

```jsx
import { NavLink } from 'react-router-dom';


export const Header = () => {

    return(
        <div>
          <NavLink to="/aside" className="nav__link">
              aside
          </NavLink>
          <NavLink to="/article" className="nav__link">
              article
          </NavLink>
        </div>
    )
}

export default Header
```

```jsx
import React, { useState } from 'react';
import { Formik, Form, Field, ErrorMessage } from 'formik';
import * as Yup from 'yup';

const validationSchema = Yup.object().shape({
  name: Yup.string()
    .required('Ім\'я є обов\'язковим полем')
    .min(2, 'Ім\'я повинно містити принаймні 2 символи'),
  email: Yup.string()
    .email('Неправильний формат Email')
    .required('Email є обов\'язковим полем'),
});

function App() {
  const [data, setData] = useState(null);

  return (
    <div className="App">
      <h1>Форма з Formik та Yup</h1>
      <Formik
        initialValues={{ name: '', email: '' }}
        validationSchema={validationSchema}
        onSubmit={values => {
          setData(values);
          console.log(values);
        }}
      >
        <Form>
          <div>
            <label htmlFor="name">Ім'я:</label>
            <Field type="text" id="name" name="name" />
            <ErrorMessage name="name" component="div" className="error" />
          </div>
          <div>
            <label htmlFor="email">Email:</label>
            <Field type="email" id="email" name="email" />
            <ErrorMessage name="email" component="div" className="error" />
          </div>
          <button type="submit">Відправити</button>
        </Form>
      </Formik>

      {data && (
        <div>
          <h2>Дані з форми:</h2>
          <p>Ім'я: {data.name}</p>
          <p>Email: {data.email}</p>
        </div>
      )}
    </div>
  );
}
```

src > components > UserProfile > UserProfile.jsx > ...

```jsx
import React, { useContext } from 'react';
import { UserContext } from '../../Context/UserContext';

export const UserProfile = () => {
  const { user } = useContext(UserContext);

  return (
    <div>
      {user ? (
        <div>
          <h2>Профіль користувача</h2>
          <p>Ім'я: {user.name}</p>
          <p>Email: {user.email}</p>
        </div>
      ) : (
        <p>Користувач не увійшов в систему.</p>
      )}
    </div>
  );
}

export default UserProfile;
```

Додаток Ж

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Комплекс мобільного керування автоматикою пропуску".
Частина 2. Програмна частина

Тип роботи: комплексна магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 99,4 % Схожість 0.6%

Аналіз звіту подібності (відмітити потрібне):

✓ Запозичення, виявлення у роботі, оформлені коректно і не містять ознак плагіату.

□ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

□ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Захарченко С.М.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Зубринська Д.Л.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Городецька О.С.
(підпис) (прізвище, ініціали)

.