Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(назва освітнього ступеня)

на тему: *Комп'ютеризована система контролю доступу із хешуванням персональних даних*

Виконав(ла): студент(ка) *IV* курсу, групи *СІс-42*
спеціальності *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

	(підпис)	Наконечний В.В. (прізвище та ініціали)
Керівник	(підпис)	Паляниця Ю.Б. (прізвище та ініціали)
Нормоконтроль	(підпис)	Тиш Є.В. (прізвище та ініціали)
Завідувач кафедри	(підпис)	Осухівська Г.М. (прізвище та ініціали)
Рецензент	(підпис)	Ясній О.П. (прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет *комп'ютерно-інформаційних систем і програмної інженерії*
(повна назва факультету)

Кафедра *комп'ютерних систем та мереж*
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Осухівська Г.М.
(підпис) (прізвище та ініціали)
« ____ » _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня *бакалавр*
(назва освітнього ступеня)

за спеціальністю *123 «Комп'ютерна інженерія»*
(шифр і назва спеціальності)

студенту *Наконечний Віталій Володимирович*
(прізвище, ім'я, по батькові)

1. Тема роботи *Комп'ютеризована система контролю доступу із хешуванням персональних даних*

Керівник роботи *Паляниця Юрій Богданович, к.т.н., ст.викладач*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «*24*» *04* *2024* року № *4/7-408*

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи *Технічне завдання*

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ
1. Аналіз технічного завдання
2.Проектна частина
3.Практична частина
4.Безпека життєдіяльності, основи охорони праці
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Структурна схема
2. Схема електрична принципова
3. Блок-схема алгоритму програми мікроконтролера
4. Блок-схема алгоритму роботи інтерфейсу

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності,	*Пилипець М.І., д.т.н., проф. каф. МТ*		
основи охорони праці			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	*Розробка та затвердження технічного завдання*	*01.02-09.02.2024*	*виконано*
2	*Аналіз технічного завдання та обґрунтування можливих*	*05.02.-11.02.2024*	*виконано*
	рішень		
3	*Розробка структурної схеми системи*	*01.06-03.06.2024*	*виконано*
4	*Розробка електричної принципової схеми, вибір*	*02.06-10.06.2024*	*виконано*
	елементної бази		
5	*Розробка програмного забезпечення для проектованої*	*10.06-16.06.2024*	*виконано*
	системи		
6	*Опрацювання питань розділу «Безпека*	*10.06-15.06.2024*	*виконано*
	життєдіяльності, основи охорони праці»		
7	*Оформлення пояснювальної записки кваліфікаційної*	*16.06-20.06.2024*	*виконано*
	роботи бакалавра		
8	*Оформлення графічної частини*	*17.06-22.06.2024*	*виконано*
9	*Попередній захист кваліфікаційної роботи бакалавра*	*14.06.2024*	*виконано*
10	*Захист кваліфікаційної роботи бакалавра*	*24.06-28.06.2024*	*виконано*

Студент _____ *Наконечний В.В.*
(підпис) (прізвище та ініціали)

Керівник роботи _____ *Паляниця Ю.Б.*
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Комп'ютеризована система контролю доступу із хешуванням персональних даних // Кваліфікаційна робота бакалавра // Наконечний Віталій Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно–інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІс-42 // Тернопіль, 2024 // с. – 154, рис. – 9, табл. – 0, кресл. – 4, додат. – 3, бібліогр. – 20.

Ключові слова: комп'ютеризована система контролю доступу, хешування, SHA-256, мікроконтролер, RFID.

Кваліфікаційну роботу бакалавра присвячено розробці комп'ютеризованої системи контролю доступу із хешуванням персональних даних, яку реалізовано на основі безконтактної технології RFID.

Після вивчення та аналізу сучасних апаратних і програмних рішень для контролю доступу у першому розділі, розроблено структурну схему комп'ютерної системи обліку робочого часу на основі RFID.

У другому розділі обґрунтовано вибір компонентів системи та детально описано процес розробки принципової електричної схеми апаратної частини, а також розроблено алгоритм функціонування апаратної складової системи та надано опис програмних функцій мікроконтролера.

У третьому розділі описано процес розробки програмного забезпечення серверної частини, що включає бекенд та фронтенд компоненти, а також використання бази даних SQLite.

У четвертому розділі розглянуто ключові аспекти безпеки життєдіяльності та охорони праці, пов'язані з розробленою системою та її експлуатацією.

ABSTRACT

Computerized access control system with personal data hashing. // bachelor's qualification work // Nakonechnyy Vitaliy Volodymyrovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Systems and Networks Department, Group CIs-42 // Ternopil, 2024 // p. - 154, fig. - 9, tab. - 0, sheets A1. - 4, appendix. – 3, bibliography – 20.

Keywords: computerized access control system, hashing, SHA-256, microcontroller, RFID.

The bachelor's qualification work is devoted to the development of a computerized access control system with personal data hashing, implemented based on contactless RFID technology.

After studying and analyzing modern hardware and software solutions for access control in the first section, a structural diagram of a computer-based time tracking system using RFID was developed.

In the second section, the choice of system components is justified and the process of developing the schematic diagram of the hardware part is described in detail. Additionally, an algorithm for the functioning of the hardware component of the system is developed and a description of the microcontroller's software functions is provided.

The third section describes the process of developing the server-side software, which includes backend and frontend components, as well as the use of an SQLite database.

The fourth section examines key aspects of life safety and occupational health related to the developed system and its operation.

ЗМІСТ

СПИСОК СКОРОЧЕНЬ

API - Application Programming Interface;

Arduino UNO - A microcontroller board;

ATmega328P - A microcontroller chip;

A/C - Alternating Current;

EEPROM - Electrically Erasable Programmable Read-Only Memory;

GPIO - General-Purpose Input/Output;

I2C - Inter-Integrated Circuit;

ISO - International Organization for Standardization;

LED - Light Emitting Diode;

MIFARE - A type of RFID card;

MFRC522 - A type of RFID reader module;

PIN - Personal Identification Number;

RFID - Radio-Frequency Identification;

SPI - Serial Peripheral Interface;

SRAM - Static Random Access Memory;

UART - Universal Asynchronous Receiver-Transmitter;

USB - Universal Serial Bus;

Wi-Fi - Wireless Fidelity.

ВСТУП

Забезпечення ефективного контролю доступу до захищених приміщень, приладів та інформації є ключовим завданням для сучасних організацій. Традиційні системи, засновані на використанні фізичних ключів та перепусток, вже давно застаріли та не можуть адекватно відповідати вимогам безпеки в умовах стрімкого розвитку інформаційних технологій. Відсутність можливості індивідуальної ідентифікації, обмежений облік дій користувачів та неоперативне реагування на спроби несанкціонованого проникнення робить такі системи вразливими до зловживань та компрометації.

Впровадження комп'ютеризованих систем контролю доступу дозволяє значно підвищити надійність та гнучкість управління доступом. Ці системи забезпечують можливість точної ідентифікації кожного користувача, детальний облік їхніх дій, а також оперативне автоматичне реагування на спроби незаконного проникнення. Така функціональність стає особливо актуальною в галузях, де збереження конфіденційності та цілісності інформації має першочергове значення, як-от державні установи, фінансові організації, підприємства з підвищеними вимогами безпеки.

Водночас, широке впровадження комп'ютеризованих систем контролю доступу супроводжується зростанням загроз, пов'язаних з несанкціонованим доступом до персональних даних користувачів. Крадіжка, підробка або незаконне розповсюдження цієї інформації може завдати значної шкоди як окремим особам, так і цілим організаціям. Дані, що зберігаються в таких системах, зазвичай включають ім'я, пароль, біометричні характеристики, ідентифікаційні номери, дані про місцезнаходження тощо. Витік або незаконне використання цих відомостей може призвести до фінансових збитків, псування репутації, а в окремих випадках - навіть до загрози життю та здоров'ю людей.

Таким чином, забезпечення надійного захисту персональних даних користувачів комп'ютеризованих систем контролю доступу стає невід'ємною складовою їх ефективного функціонування. Впровадження сучасних технологій хешування, які дозволяють перетворювати відкриті дані у складні, нерозбірливі рядки символів, є ключовим інструментом забезпечення конфіденційності, цілісності та доступності sensitive інформації. Ефективне застосування таких технологій є критично важливим для мінімізації ризиків, пов'язаних з несанкціонованим доступом до персональних даних в комп'ютеризованих системах контролю доступу.

Отже, розробка та впровадження комп'ютеризованих систем контролю доступу, які забезпечують надійний захист персональних даних користувачів з використанням технологій хешування, є надзвичайно актуальною та важливою проблемою сучасності. Вирішення цього завдання дозволить значно підвищити рівень безпеки управління доступом до критичних ресурсів, мінімізувати ризики витоку конфіденційної інформації та захистити права й інтереси як окремих осіб, так і організацій в цілому.

РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ ТА ОСОБЛИВОСТЕЙ ФУНКЦІОНУВАННЯ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ КОНТРОЛЮ ДОСТУПУ ІЗ ХЕШУВАННЯМ ПЕРСОНАЛЬНИХ ДАНИХ

1.1 Актуальність проблеми комп'ютеризованих систем контролю доступу

Недоліки традиційних систем контролю доступу. Традиційні системи контролю доступу [1,2], засновані на використанні фізичних ключів та перепусток, мають низку суттєвих недоліків, які значно знижують їх ефективність та надійність у сучасних умовах. Серед основних проблем таких систем можна виділити наступні:

Обмежена можливість ідентифікації користувачів. Традиційні системи не забезпечують точної ідентифікації конкретної особи, яка отримує доступ. Ключі або перепустки можуть бути передані іншим людям, а в разі їх втрати — використані зловмисниками. Це може призвести до незаконного доступу до критичних ресурсів, що може бути особливо шкідливим для організацій, які обробляють конфіденційну інформацію.

Відсутність деталізованого обліку дій користувачів. Такі системи не дозволяють відстежувати, коли саме і хто саме отримував доступ до певних зон або ресурсів. Це ускладнює розслідування інцидентів та порушень безпеки, що може призвести до затримок у вирішенні кризових ситуацій.

Неоперативне реагування на спроби несанкціонованого доступу. У разі втрати ключа або компрометації перепустки система не може швидко заблокувати доступ, оскільки потрібен тривалий час для заміни замків або

анулювання пропусків. Це може дати зловмисникам достатньо часу для несанкціонованого доступу до критичних ресурсів.

Обмежена масштабованість та гнучкість. Зі зростанням кількості користувачів та зон контролю доступу ускладнюється управління традиційними системами, збільшується кількість ключів та перепусток, що потребує додаткових ресурсів та зусиль для забезпечення безпеки. Це може бути особливо критичним для організацій, які швидко зростають або мають велику кількість працівників.

Відсутність централізованого управління. Кожна точка доступу контролюється окремо, що ускладнює моніторинг та аналіз подій безпеки в масштабах всієї системи. Це може призвести до незапланованих затримок та витрат на коригування систем.

Високі витрати на встановлення та обслуговування. Необхідність фізичної заміни замків, виготовлення дублікатів ключів, друку та розповсюдження перепусток вимагає значних фінансових та трудових витрат. Це може бути особливо важливим для організацій, які мають обмежені ресурси.

Наведені недоліки традиційних систем контролю доступу стають особливо критичними в умовах зростаючих вимог до безпеки в різних галузях, таких як державні установи, фінансові організації, об'єкти критичної інфраструктури тощо. Саме тому впровадження сучасних комп'ютеризованих систем контролю доступу стає невід'ємною складовою забезпечення належного рівня захисту [3,4].

Переваги комп'ютеризованих систем. Комп'ютеризовані системи контролю доступу (КСКД) надають безліч переваг, які роблять їх більш ефективними та надійними порівняно з традиційними системами. Нижче наведені основні переваги КСКД:

1. Точна ідентифікація користувачів: Комп'ютеризовані системи можуть забезпечити точну ідентифікацію користувачів, що дозволяє відстежувати, хто

отримує доступ до яких ресурсів та коли. Це значно зменшує ризики несанкціонованого доступу та дозволяє оперативніше реагувати на спроби зловмисників.

2. Деталізований облік дій користувачів: КСКД дозволяють відстежувати всі дії користувачів, що дозволяє оперативно реагувати на порушення безпеки та виявляти потенційні загрози раніше.

3. Оперативне реагування на спроби несанкціонованого доступу: Комп'ютеризовані системи можуть швидко заблокувати доступ у разі спроби несанкціонованого доступу, що дозволяє оперативно реагувати на загрози та мінімізувати ризики.

4. Масштабованість та гнучкість: КСКД можуть бути легко масштабовані для покриття великої кількості користувачів та зон доступу, що дозволяє адаптуватися до змінних потреб організації.

5. Централізоване управління: Комп'ютеризовані системи дозволяють централізовано управляти доступом до різних зон та ресурсів, що дозволяє оперативно реагувати на загрози та виявляти потенційні проблеми раніше.

6. Низькі витрати на обслуговування: Комп'ютеризовані системи потребують менше ресурсів та зусиль для обслуговування, що дозволяє економити кошти та ресурси.

7. Безпека та конфіденційність: КСКД забезпечують надійний захист персональних даних користувачів, що дозволяє забезпечити конфіденційність та цілісність інформації.

8. Мониторинг та аналіз подій безпеки: Комп'ютеризовані системи дозволяють моніторити та аналізувати події безпеки, що дозволяє оперативно реагувати на загрози та виявляти потенційні проблеми раніше.

9. Використання різних технологій: КСКД можуть використовувати різні технології, такі як біометрична ідентифікація, електронні картки доступу та гібридні системи, що дозволяє вибрати найефективніший підхід для конкретної організації.

10. Мінімізація ризиків: Комп'ютеризовані системи можуть мінімізувати ризики несанкціонованого доступу, крадіжки та підробки даних, що дозволяє забезпечити безпеку критичних ресурсів.

11. Регулярне оновлення та підтримка: Комп'ютеризовані системи можуть бути регулярно оновлювані та підтримувані, що дозволяє забезпечити їх стабільну роботу та захист від нових загроз.

12. Використання аналітичних даних: Комп'ютеризовані системи можуть використовувати аналітичні дані для виявлення потенційних загроз та покращення безпеки.

13. Деталізація доступу: КСКД можуть забезпечити деталізацію доступу до різних ресурсів, що дозволяє оперативно реагувати на спроби несанкціонованого доступу.

14. Використання різних типів аутентифікації: Комп'ютеризовані системи можуть використовувати різні типи аутентифікації, такі як паролі, біометричні дані та електронні картки, що дозволяє забезпечити вищу безпеку.

15. Мінімізація витрат на обслуговування: Комп'ютеризовані системи можуть мінімізувати витрати на обслуговування, оскільки вони потребують менше ресурсів та зусиль для підтримки.

Важливість для забезпечення безпеки в різних галузях. Важливість комп'ютеризованих систем контролю доступу для забезпечення безпеки в різних галузях полягає в тому, що вони надають безліч переваг, які роблять їх більш ефективними та надійними порівняно з традиційними системами.

Нижче наведені основні переваги комп'ютеризованих систем контролю доступу:

1. Точна ідентифікація користувачів: Комп'ютеризовані системи можуть забезпечити точну ідентифікацію користувачів, що дозволяє відстежувати, хто отримує доступ до яких ресурсів та коли. Це значно зменшує ризики несанкціонованого доступу та дозволяє оперативніше реагувати на спроби зловмисників.

2. Деталізований облік дій користувачів: КСКД дозволяють відстежувати всі дії користувачів, що дозволяє оперативно реагувати на порушення безпеки та виявляти потенційні загрози раніше.

3. Оперативне реагування на спроби несанкціонованого доступу: Комп'ютеризовані системи можуть швидко заблокувати доступ у разі спроби несанкціонованого доступу, що дозволяє оперативно реагувати на загрози та мінімізувати ризики.

4. Масштабованість та гнучкість: КСКД можуть бути легко масштабовані для покриття великої кількості користувачів та зон доступу, що дозволяє адаптуватися до змінних потреб організації.

5. Централізоване управління: Комп'ютеризовані системи дозволяють централізовано управляти доступом до різних зон та ресурсів, що дозволяє оперативно реагувати на загрози та виявляти потенційні проблеми раніше.

6. Низькі витрати на обслуговування: Комп'ютеризовані системи потребують менше ресурсів та зусиль для обслуговування, що дозволяє економити кошти та ресурси.

7. Безпека та конфіденційність: КСКД забезпечують надійний захист персональних даних користувачів, що дозволяє забезпечити конфіденційність та цілісність інформації.

8. Мониторинг та аналіз подій безпеки: Комп'ютеризовані системи дозволяють моніторити та аналізувати події безпеки, що дозволяє оперативно реагувати на загрози та виявляти потенційні проблеми раніше.

9. Використання різних технологій: КСКД можуть використовувати різні технології, такі як біометрична ідентифікація, електронні картки доступу та гібридні системи, що дозволяє вибрати найефективніший підхід для конкретної організації.

10. Мінімізація ризиків: Комп'ютеризовані системи можуть мінімізувати ризики несанкціонованого доступу, крадіжки та підробки даних, що дозволяє забезпечити безпеку критичних ресурсів.

11. Регулярне оновлення та підтримка: Комп'ютеризовані системи можуть бути регулярно оновлювані та підтримувані, що дозволяє забезпечити їх стабільну роботу та захист від нових загроз.

12. Використання аналітичних даних: Комп'ютеризовані системи можуть використовувати аналітичні дані для виявлення потенційних загроз та покращення безпеки.

13. Деталізація доступу: КСКД можуть забезпечити деталізацію доступу до різних ресурсів, що дозволяє оперативно реагувати на спроби несанкціонованого доступу.

14. Використання різних типів аутентифікації: Комп'ютеризовані системи можуть використовувати різні типи аутентифікації, такі як паролі, біометричні дані та електронні картки, що дозволяє забезпечити вищу безпеку.

15. Мінімізація витрат на обслуговування: Комп'ютеризовані системи можуть мінімізувати витрати на обслуговування, оскільки вони потребують менше ресурсів та зусиль для підтримки.

Таким чином, комп'ютеризовані системи контролю доступу є критично важливими для забезпечення безпеки в різних галузях, таких як державні установи, фінансові організації, об'єкти критичної інфраструктури та інші, де збереження конфіденційності та цілісності інформації має першочергове значення.

Комп'ютеризовані системи контролю доступу дозволяють оперативно реагувати на загрози, що значно зменшує ризики несанкціонованого доступу та крадіжки даних. Вони можуть бути легко масштабовані для покриття великої кількості користувачів та зон доступу, що дозволяє адаптуватися до змінних потреб організації.

Централізоване управління доступом до різних зон та ресурсів дозволяє оперативно реагувати на загрози та виявляти потенційні проблеми раніше. Комп'ютеризовані системи контролю доступу також забезпечують надійний

захист персональних даних користувачів, що дозволяє забезпечити конфіденційність та цілісність інформації.

Моніторинг та аналіз подій безпеки дозволяють оперативно реагувати на загрози та виявляти потенційні проблеми раніше. Комп'ютеризовані системи можуть використовувати різні технології, такі як біометрична ідентифікація, електронні картки доступу та гібридні системи, що дозволяє вибрати найефективніший підхід для конкретної організації.

Деталізація доступу до різних ресурсів дозволяє оперативно реагувати на спроби несанкціонованого доступу. Комп'ютеризовані системи можуть використовувати різні типи аутентифікації, такі як паролі, біометричні дані та електронні картки, що дозволяє забезпечити вищу безпеку.

Мінімізація витрат на обслуговування комп'ютеризованих систем контролю доступу дозволяє економити кошти та ресурси. В цілому, комп'ютеризовані системи контролю доступу є критично важливими для забезпечення безпеки в різних галузях, де збереження конфіденційності та цілісності інформації має першочергове значення.

1.2 Необхідність захисту персональних даних за допомогою хешування

Необхідність захисту персональних даних за допомогою хешування є критично важливим аспектом сучасних комп'ютеризованих систем контролю доступу. У світі, де інформаційні технології стрімко розвиваються, а кількість кіберзагроз постійно зростає, забезпечення конфіденційності, цілісності та доступності даних стає першочерговим завданням для будь-якої організації. Хешування як метод криптографічного захисту інформації відіграє ключову роль у вирішенні цієї проблеми. Сутність хешування полягає у перетворенні вхідних даних довільної довжини у вихідний бітовий рядок фіксованої довжини. Цей процес є незворотним, що означає неможливість відновлення початкових даних з хеш-значення. Така властивість робить хешування

надзвичайно ефективним інструментом для захисту персональних даних користувачів систем контролю доступу. Використання хешування дозволяє зберігати не самі конфіденційні дані, а лише їх хеш-значення, що суттєво знижує ризики у разі несанкціонованого доступу до бази даних. Наприклад, замість зберігання паролів користувачів у відкритому вигляді, система зберігає лише їх хеш-значення. При спробі автентифікації введений користувачем пароль хешується і порівнюється з збереженим хеш-значенням. Це забезпечує високий рівень захисту навіть у випадку компрометації бази даних. Крім того, хешування дозволяє ефективно перевіряти цілісність даних. Будь-яка зміна вхідних даних призводить до зміни їх хеш-значення, що дозволяє легко виявити несанкціоновані модифікації. Це особливо важливо для систем контролю доступу, де цілісність інформації про права доступу та історію операцій має критичне значення для безпеки об'єкта, що охороняється. Застосування хешування також сприяє підвищенню продуктивності системи. Оскільки хеш-значення мають фіксовану довжину, незалежно від розміру вхідних даних, це оптимізує процеси пошуку та порівняння інформації в базі даних. Це особливо актуально для великих систем з багатьма користувачами та високою інтенсивністю операцій. Важливо відзначити, що ефективність хешування значною мірою залежить від вибору відповідного алгоритму. Сучасні криптографічні хеш-функції, такі як SHA-256 або SHA-3, забезпечують високий рівень захисту від колізій (ситуацій, коли різні вхідні дані дають однакове хеш-значення) та інших видів атак. При цьому постійний розвиток обчислювальних технологій вимагає періодичного оновлення використовуваних алгоритмів для підтримки належного рівня безпеки. Впровадження хешування в системи контролю доступу також сприяє дотриманню вимог законодавства щодо захисту персональних даних, таких як GDPR в Європейському Союзі або аналогічні нормативні акти в інших країнах. Це не тільки знижує ризики юридичних санкцій, але й підвищує довіру користувачів до системи, що є важливим фактором для репутації

організації. Хешування також відіграє важливу роль у реалізації принципу мінімізації даних, який є одним з ключових в сучасних підходах до захисту приватності. Зберігаючи лише хеш-значення замість повних персональних даних, організації можуть суттєво зменшити обсяг чутливої інформації, що потребує захисту. Це не тільки знижує ризики у разі витоку даних, але й спрощує процеси управління інформаційною безпекою. Крім того, використання хешування відкриває можливості для впровадження додаткових механізмів безпеки, таких як сольовання (додавання випадкових даних до вхідної інформації перед хешуванням) або багаторазове хешування. Ці методи ще більше підвищують стійкість системи до різних видів атак, включаючи атаки з використанням райдужних таблиць або брутфорс. У контексті систем контролю доступу хешування також може бути використане для створення унікальних ідентифікаторів користувачів або пристроїв, що підвищує загальну безпеку системи та ускладнює можливість підробки або клонування засобів доступу. Таким чином, необхідність захисту персональних даних за допомогою хешування в комп'ютеризованих системах контролю доступу є беззаперечною. Це не просто технічний аспект, а фундаментальний принцип, який забезпечує конфіденційність, цілісність та доступність критично важливої інформації. Впровадження ефективних методів хешування дозволяє організаціям не тільки захистити дані своїх користувачів, але й побудувати надійну та довірену систему контролю доступу, здатну протистояти сучасним кіберзагрозам та відповідати найвищим стандартам інформаційної безпеки.

1.3 Актуальність проблеми розробки захищених комп'ютеризованих систем

Актуальність проблеми розробки захищених комп'ютеризованих систем контролю доступу є надзвичайно високою в сучасному світі, де інформаційна безпека стає ключовим фактором успіху та стабільності організацій різних

масштабів та галузей. Ця проблема набуває особливого значення в контексті постійно зростаючих кіберзагроз та ускладнення методів несанкціонованого доступу до інформації.

Підсумовуючи важливість проблеми, слід зазначити, що захищені комп'ютеризовані системи контролю доступу є критичним елементом інфраструктури безпеки сучасних підприємств, установ та організацій. Вони не лише забезпечують фізичний контроль доступу до приміщень та об'єктів, але й відіграють ключову роль у захисті інформаційних активів, включаючи конфіденційні дані співробітників, клієнтів та партнерів. В епоху цифрової трансформації, коли більшість бізнес-процесів переноситься в онлайн-середовище, надійна система контролю доступу стає першою лінією оборони проти широкого спектру загроз [5-7] - від промислового шпіонажу до кібертероризму.

Особливу актуальність проблема набуває у світлі зростаючих вимог до захисту персональних даних, що регулюються такими нормативними актами, як GDPR в Європейському Союзі та аналогічними законами в інших країнах. Організації, які не зможуть забезпечити належний рівень захисту даних своїх користувачів, ризикують не лише втратити довіру клієнтів, але й зіткнутися з серйозними юридичними та фінансовими наслідками.

Перспективи вирішення проблеми розробки захищених комп'ютеризованих систем контролю доступу є багатообіцяючими, але вимагають комплексного підходу та постійного вдосконалення. Одним з ключових напрямків є інтеграція передових технологій, таких як штучний інтелект та машинне навчання, для підвищення точності ідентифікації та виявлення аномальної поведінки. Використання блокчейн-технологій може забезпечити непорушність та прозорість журналів доступу, що критично важливо для аудиту безпеки та розслідування інцидентів.

Потенційні переваги від впровадження сучасних захищених систем контролю доступу є значними. По-перше, це суттєве підвищення рівня

безпеки об'єктів та інформації, що зберігається та обробляється організацією. Це включає не лише запобігання несанкціонованому фізичному доступу, але й захист від витоку даних та кіберзагроз. По-друге, такі системи можуть значно підвищити ефективність управління доступом, автоматизуючи багато процесів та мінімізуючи людський фактор. Це призводить до оптимізації робочих процесів та зниження операційних витрат.

Крім того, розвиток захищених комп'ютеризованих систем контролю доступу відкриває нові можливості для бізнесу. Наприклад, інтеграція з системами управління персоналом дозволяє автоматично регулювати права доступу відповідно до зміни посади або статусу співробітника. А використання біометричних технологій не тільки підвищує безпеку, але й може покращити користувацький досвід, роблячи процес аутентифікації більш зручним та швидким.

Важливо також відзначити, що розробка таких систем стимулює інновації в суміжних областях, таких як криптографія, біометрія та аналіз даних. Це створює нові робочі місця та сприяє розвитку високотехнологічних галузей економіки.

Однак, разом з перевагами, розвиток захищених комп'ютеризованих систем контролю доступу ставить перед суспільством і ряд етичних питань, зокрема, щодо балансу між безпекою та приватністю. Вирішення цих питань вимагає не лише технологічних, але й правових та соціальних інновацій.

Таким чином, актуальність проблеми розробки захищених комп'ютеризованих систем контролю доступу не обмежується лише технологічним аспектом. Це комплексне завдання, що охоплює питання безпеки, ефективності, етики та регулювання. Успішне вирішення цієї проблеми має потенціал не лише підвищити рівень захищеності організацій та індивідів, але й стати каталізатором інновацій та економічного розвитку в епоху цифрової трансформації.

1.4 Огляд ринку існуючих систем контролю доступу

Системи доступу є критичним аспектом сучасної безпеки, забезпечуючи, що лише авторизовані особи мають доступ до конкретних зон або ресурсів. Декілька репутованих виробників пропонують інноваційні розв'язки для задоволення різних потреб безпеки.

ButterflyMX, заснована в 2013 році [8], є провідним постачальником комплексних рішень для керування доступом для широкомасштабних застосувань у будівлях. Їхні інноваційні системи пропонують широкий спектр передових функцій та можливостей, які задовольняють різноманітні потреби житлових, комерційних та багатоквартирних будівель.

Один з ключових акцентів систем керування доступом ButterflyMX полягає в їхній безшовній інтеграції з широкомасштабними розв'язками. Системи можуть бути інтегровані з інтеркомами, відеоспостереженням та мобільними додатками, забезпечуючи єдиний та зручний досвід для мешканців будівель, орендарів та менеджерів будівель.

Серцем технології керування доступом ButterflyMX є їхня хмарна платформа, яка дозволяє дистанційному керуванню та моніторингу точок доступу в реальному часі. Ця хмарна підхід дозволяє легко конфігурувати, керувати користувачами та модифікувати керування доступом з будь-якого місця, надавши менеджерам будівель покращений контроль та видимість над їхньою інфраструктурою безпеки.

Системи керування доступом ButterflyMX характеризуються рядом передових компонентів апаратного забезпечення, включаючи високоякісні відеоінтеркоми, читачі карток та сканери біометричних даних. Ці пристрої використовують передові технології, такі як розпізнавання облич, сканування QR-кодів та RFID, щоб забезпечити кілька методів автентифікації для безпечного керування доступом.

Мобільні додатки компанії, доступні для iOS та Android, надають мешканцям та орендарям можливість дистанційно керувати доступом до їхніх будівель та квартир. Користувачі можуть надати тимчасовий доступ гостьовим, отримувати сповіщення про приїзд гостей та навіть відкривати двері або ворота з їхніх смартфонів, покращуючи загальну зручність та безпеку будівлі.

Системи керування доступом ButterflyMX спроектовані з урахуванням масштабованості, дозволяючи легко розширювати їх за потребами будівлі. Модульна архітектура систем дозволяє додавати нові точки доступу, інтегрувати з додатковими пристроями безпеки та розширювати загальну мережу керування доступом.

Що стосується безпеки даних, розв'язки керування доступом ButterflyMX оснащені потужними протоколами шифрування та автентифікації, забезпечуючи конфіденційність та цілісність чутливих даних про доступ. Хмарна платформа також регулярно проходить аудити безпеки та оновлення, щоб підтримувати найвищі рівні захисту від кіберзагроз.

Системи керування доступом ButterflyMX відомі своєю зручною інтерфейсом, як для мешканців/орендарів, так і для менеджерів будівель. Інтуїтивна веб-консоль керування та мобільні додатки спрощують адміністрування дозволів на доступ, керування гостьовим та звітність в реальному часі, спрощуючи щоденні операції команд будівельного менеджменту.

Зобов'язання ButterflyMX щодо інновацій очевидне в їхніх постійних зусиллях щодо розвитку продуктів. Компанія регулярно представляє нові функції та покращення до своїх систем керування доступом, забезпечуючи клієнтам можливість використовувати останні досягнення в галузі безпеки технологій, щоб задовольнити еволюційні потреби їхніх будівель.

Загалом, розв'язки керування доступом ButterflyMX пропонують комплексний, хмарний та зручний підхід до керування безпекою будівель та

громад. З їхніми передовими компонентами апаратного забезпечення, інтегрованими програмними розв'язками та мобільними функціями, ButterflyMX надає власникам та менеджерам будівель потужну та масштабовану систему керування доступом, яка покращує загальну безпеку та зручність їхніх будівель та громад.

Brivo – одна з провідних компаній у сфері хмарних рішень для контролю доступу [9]. Заснована у 2002 році, Brivo зарекомендувала себе як надійний та інноваційний виробник надійних систем контролю доступу. Ключовою перевагою Brivo є її хмарна платформа, яка дозволяє клієнтам легко керувати та масштабувати системи контролю доступу без необхідності в локальному сервері або складному програмному забезпеченні.

Одним із флагманських продуктів Brivo є система Brivo OnAir. Ця хмарна платформа надає широкі можливості для контролю доступу, включаючи управління користувачами, створення розкладів, відстеження входу/виходу та аналітику. Завдяки хмарному підходу, Brivo OnAir усуває необхідність в обслуговуванні локального обладнання, що суттєво знижує витрати та зусилля на систему контролю доступу.

Технологічно, Brivo OnAir базується на надійній та масштабованій хмарній інфраструктурі AWS. Завдяки цьому, система забезпечує високу доступність, безперебійність роботи та безпеку даних. Клієнти можуть дистанційно отримувати доступ до системи через веб-інтерфейс або мобільний додаток, що забезпечує зручність управління.

Brivo пропонує широкий спектр зчитувачів, контролерів та інших периферійних пристроїв, сумісних з платформою Brivo OnAir. Зчитувачі підтримують різні технології, такі як карти доступу, мобільні додатки та біометрія. Контролери забезпечують надійне та масштабоване керування доступом, підтримуючи високу пропускну здатність та резервування.

Для інтеграції з іншими системами, Brivo OnAir має потужні API, які дозволяють легко підключати додаткові функції, такі як відеоспостереження,

управління відвідувачами або системи автоматизації будівель. Це дає змогу клієнтам створювати комплексні рішення, що відповідають їхнім унікальним вимогам.

Безпека є ключовим пріоритетом для Brivo. Платформа Brivo OnAir використовує передові технології шифрування та аутентифікації, щоб гарантувати конфіденційність та захист даних клієнтів. Система регулярно проходить аудити безпеки та відповідає високим стандартам галузі.

Крім того, Brivo приділяє велику увагу зручності користувачів. Веб-інтерфейс та мобільні додатки мають інтуїтивно зрозумілий дизайн, що полегшує повсякденне управління системою контролю доступу. Також Brivo пропонує всебічні навчальні матеріали та підтримку, щоб допомогти клієнтам максимально ефективно використовувати систему.

Технологія Brivo постійно удосконалюється, оскільки компанія регулярно впроваджує нові функції та покращення. Наприклад, було додано підтримку мобільного доступу, інтеграцію з відеоспостереженням та аналітику поведінки користувачів. Це дозволяє клієнтам Brivo постійно отримувати інноваційні можливості для управління контролем доступу.

Масштабованість та гнучкість – ще одні ключові переваги Brivo OnAir. Система здатна обслуговувати як невеликі офіси, так і великі підприємства з тисячами користувачів та дверей. Клієнти можуть легко розширювати та налаштовувати систему відповідно до зростаючих потреб.

У підсумку, Brivo виділяється як один із лідерів у сфері хмарних рішень контролю доступу. Завдяки передовим технологіям, масштабованості та зручності використання, платформа Brivo OnAir надає клієнтам надійне та гнучке управління доступом до їхніх об'єктів. Постійні інновації та увага до безпеки роблять Brivo привабливим вибором для організацій, які шукають сучасне та ефективне рішення для контролю доступу.

Genetec, заснована в 1997 році, є визнаним гравцем у галузі контролю доступу [10]. Їхня флагманська система, відома як Synergis, пропонує широкий спектр високопродуктивних та інноваційних рішень.

Система Synergis від Genetec складається з кількох ключових елементів. Контролер доступу Synergis є серцем системи, забезпечуючи централізоване управління дозволами та подіями. Він може керувати до 32 дверима та 100 000 ідентифікаційними картками на контролер. Контролер Synergis має внутрішній буфер пам'яті, який дозволяє йому продовжувати роботу навіть у разі втрати підключення до сервера.

Зчитувачі доступу Genetec розроблені для ідеальної інтеграції з системою Synergis. Вони підтримують широкий спектр технологій ідентифікації, у тому числі смарт-карти, електронні ключі та смартфони через технологію Bluetooth. Ці зчитувачі забезпечують швидке та надійне зчитування даних ідентифікації, гарантуючи плавний та безпечний контроль доступу.

Програмне забезпечення Security Center, розроблене Genetec, є уніфікованою платформою для управління всією системою контролю доступу Synergis. Цей інтуїтивно зрозумілий та зручний інтерфейс пропонує безліч передових функцій, таких як управління пропусками, моніторинг подій у реальному часі, детальні звіти та інтеграція з іншими системами безпеки, такими як відеоспостереження.

Відкрита архітектура Synergis дозволяє легку інтеграцію з широким колом сторонніх систем, таких як контролери доступу інших виробників, системи управління будівлями, системи охоронної сигналізації тощо. Ця інтероперабельність забезпечує високу гнучкість у створенні індивідуальних рішень безпеки.

Synergis також пропонує опції резервування та високої доступності, гарантуючи безперервність обслуговування в разі збоїв чи відмов [11]. Функції

резервного копіювання та відновлення дозволяють легко зберігати конфігурацію та дані системи.

Лінійка продуктів Synergis включає також спеціальні версії, призначені для задоволення конкретних потреб, наприклад, Synergis Cloud Link для хмарних додатків контролю доступу або Synergis Appliance для спрощеного розгортання на місцях.

Завдяки своєму інноваційному підходу, надійності та гнучкості система Synergis від Genetec стає еталоном у галузі контролю доступу. Її передові можливості ідентифікації, управління дозволами та інтероперабельності роблять її вибором номер один для найбільш складних встановлень.

Genetec також забезпечує своїх клієнтів якісним сервісом та технічною підтримкою, гарантуючи успішне впровадження та ефективне обслуговування їхніх систем контролю доступу Synergis.

У цілому, Genetec виділяється своїм експертизою та інноваціями у сфері контролю доступу, пропонуючи підприємствам та організаціям надійні та масштабовані рішення для забезпечення безпеки їхніх будівель та активів.

Avigilon, провідний постачальник рішень для контролю доступу, пропонує широкий спектр передових продуктів та технологій для задоволення різних потреб безпеки бізнесу та організацій. Зосереджуючись на інноваціях та надійності, системи контролю доступу Avigilon розроблені для забезпечення вищого рівня безпеки, зручності та безшовної інтеграції.

Один з флагманських продуктів у портфоліо систем контролю доступу Avigilon - це система менеджменту контролю доступу Avigilon (ACM). Система ACM є потужним та масштабованим рішенням, яке може бути налаштоване під конкретні вимоги будь-якої установи, від малих офісів до великих підприємств. Система має користувацький інтерфейс, що дозволяє адміністраторам легко керувати та моніторити діяльність контролю доступу з централізованої платформи.

Серцем системи Avigilon ACM є її передова апаратура контролю доступу. Система використовує широкий спектр високоякісних контролерів дверей та читачів, розроблених для внутрішніх та зовнішніх середовищ. Ці пристрої мають стійку конструкцію та оснащені останніми технологіями шифрування для забезпечення цілісності даних доступу та запобігання несанкціонованому доступу.

Система Avigilon ACM також пропонує комплексний набір програмних інструментів та застосунків. Потужне програмне забезпечення для керування дозволяє адміністраторам легко налаштовувати та налаштовувати політики доступу, моніторити події системи та генерувати детальні звіти. Крім того, мобільний додаток дозволяє користувачам віддалено керувати функціями контролю доступу, такими як надання або скасування дозволів доступу, з їхніх смартфонів або планшетів.

Іншою ключовою функцією системи Avigilon ACM є її передові можливості інтеграції. Система безшовно інтегрується з широким спектром систем безпеки третіх сторін, включаючи системи відеоспостереження, виявлення вторгнення та управління відвідувачами. Ця інтеграція дозволяє досягти єдиного та структурованого підходу до безпеки, дозволяючи організаціям досягти вищого рівня ситуаційної свідомості та оперативної ефективності.

Рішення Avigilon для контролю доступу також пропонують передові можливості біометричної автентифікації. Читачі біометричних даних компанії, які підтримують технології, такі як розпізнавання відбитків пальців та розпізнавання обличчя, забезпечують додатковий рівень безпеки, гарантуючи, що лише авторизовані особи можуть отримати доступ до обмежених зон.

Крім системи Avigilon ACM, компанія також пропонує широкий спектр спеціалізованих продуктів для контролю доступу, таких як пристрій для контролю доступу Avigilon (ACA) та система управління відвідувачами

Avigilon (VMS). ACA - це компактне та економічне рішення, розроблене для менших установ, тоді як VMS спрощує процес управління відвідувачами, автоматизуючи реєстрацію відвідувачів, друк бейджів та контроль доступу.

Зобов'язання Avigilon щодо інновацій демонструється також розвитком передових технологій контролю доступу, таких як мобільні рішення для доступу. Ці рішення дозволяють користувачам безпечно отримувати доступ до установ, використовуючи свої смартфони або інші мобільні пристрої, забезпечуючи зручну та гнучку альтернативу традиційним методам контролю доступу.

Підґрунтям пропозицій Avigilon для контролю доступу є сильна увага до безпеки та відповідності. Продукти компанії розроблені для відповідності останнім промисловим стандартам та регуляторним вимогам, забезпечуючи, що організації можуть підтримувати високий рівень безпеки та дотримуватися відповідних вимог до конфіденційності даних та контролю доступу.

Загалом, рішення Avigilon для контролю доступу є свідченням зобов'язання компанії надавати інноваційні, надійні та зручні технології безпеки. З широким спектром продуктів та зосередженням на інтеграції та передових функціях, Avigilon добре підготовлена до задоволення еволюючих потреб безпеки бізнесу та організацій у різних галузях.

Bosch - це один із провідних виробників систем контролю доступу на ринку. Компанія славиться своїми високоякісними та надійними продуктами, які застосовуються в широкому спектрі об'єктів - від малих офісів до великих корпоративних комплексів.

Система контролю доступу Bosch побудована на базі передових технологій, що забезпечують ефективне та надійне управління доступом. Основу системи становить контролер доступу - пристрій, який обробляє та аналізує всі дані про доступ, отримані від зчитувачів і пристроїв.

Контролери Bosch відрізняються потужною продуктивністю, вони здатні обробляти великі обсяги інформації про доступ, забезпечуючи миттєве

реагування на події. Завдяки підтримці сучасних стандартів безпеки, таких як AES-128 шифрування, контролери гарантують надійний захист даних і цілісність системи.

Зчитувачі ідентифікаторів, що входять до складу системи контролю доступу Bosch, представлені різноманітними моделями, призначеними для використання в різних умовах. Вони відрізняються типом зчитування (контактні, безконтактні, біометричні), дизайном, ступенем захисту від впливу навколишнього середовища. Це дає змогу підібрати оптимальне рішення для кожного об'єкта.

Ідентифікатори, які використовуються в системах Bosch, також відрізняються широким вибором - від традиційних карток до сучасних смартфонів із підтримкою мобільного доступу. Це дозволяє організувати гнучку систему контролю доступу, адаптовану під потреби користувачів.

Особливістю системи контролю доступу Bosch є її масштабованість. Контролери підтримують підключення великої кількості зчитувачів і пристроїв, що дає змогу створювати комплексні системи для організацій будь-якого розміру. Це забезпечується як апаратними, так і програмними засобами, які дозволяють легко розширювати та модернізувати систему.

Програмне забезпечення Bosch для контролю доступу відрізняється зручним і інтуїтивно зрозумілим інтерфейсом. Воно дозволяє централізовано управляти всіма аспектами системи - від налаштування доступу до аналізу подій. Інтеграція з іншими системами, такими як відеоспостереження або пожежна сигналізація, підвищує ефективність управління безпекою об'єкта.

Широкий спектр додаткових функцій, таких як облік робочого часу, інтеграція з системами обліку персоналу, статистика та аналітика, дозволяють системам Bosch вирішувати не лише завдання контролю доступу, а й комплексно управляти безпекою та ефективністю бізнес-процесів.

Підсумовуючи, системи контролю доступу Bosch є надійним, гнучким і масштабованим рішенням, здатним задовольнити потреби організацій різного

розміру та сфери діяльності. Поєднання передових технологій, якості виконання та широкого функціоналу робить продукти Bosch одними з найкращих на ринку систем контролю доступу.

1.5 Аналіз технічного завдання на комп'ютеризовану систему контролю доступу із хешуванням персональних даних

Комп'ютеризована система контролю доступу із хешуванням персональних даних призначена для забезпечення високого рівня безпеки та захисту конфіденційної інформації в організаціях різного масштабу та сфери діяльності. Система має на меті не лише контроль фізичного доступу до приміщень, але й захист персональних даних користувачів від несанкціонованого доступу та можливих витоків.

До складу системи повинні входити як апаратні, так і програмні компоненти. Апаратна частина має включати сучасні засоби ідентифікації, такі як зчитувачі RFID-карток. Програмна складова повинна забезпечувати обробку та зберігання даних з використанням сучасних алгоритмів хешування для захисту персональної інформації.

У рамках кваліфікаційної роботи необхідно провести детальний аналіз існуючих технологій контролю доступу та методів захисту даних. Особливу увагу слід приділити вивченню та порівнянню різних алгоритмів хешування, таких як SHA-256, bcrypt, Argon2, з метою вибору найбільш оптимального для даної системи. Важливо також розглянути можливість використання додаткових методів підвищення безпеки, таких як сольування хешів та багаторазове хешування.

Система повинна включати наступні ключові компоненти:

1. Модуль ідентифікації користувачів, який підтримує різні методи аутентифікації (RFID-картки, PIN-коди).

2. Базу даних для зберігання інформації про користувачів, їх права доступу та історію входів/виходів. При цьому всі персональні дані повинні зберігатися виключно у вигляді хеш-значень.

3. Серверну частину для обробки запитів на доступ, перевірки прав та логування подій.

4. Клієнтську частину з інтуїтивно зрозумілим інтерфейсом для адміністраторів системи, що дозволяє управляти правами доступу, генерувати звіти та налаштовувати параметри безпеки.

У процесі розробки необхідно приділити особливу увагу забезпеченню високої продуктивності системи, здатної обробляти велику кількість запитів на доступ в режимі реального часу. Також важливо передбачити можливість масштабування системи для роботи з зростаючою кількістю користувачів та об'єктів доступу.

Окремим завданням є розробка механізмів резервного копіювання та відновлення даних, а також створення процедур регулярного аудиту безпеки системи. Необхідно також врахувати вимоги до сумісності з існуючими корпоративними системами та можливість інтеграції з іншими засобами безпеки (наприклад, системами відеоспостереження).

Важливим аспектом є забезпечення відповідності системи вимогам законодавства щодо захисту персональних даних, таким як GDPR в Європейському Союзі або аналогічним нормам в інших країнах.

Доцільність створення такої комп'ютеризованої системи контролю доступу з хешуванням персональних даних полягає у значному підвищенні рівня інформаційної безпеки організації, мінімізації ризиків, пов'язаних з витоком конфіденційної інформації, та забезпеченні відповідності сучасним стандартам захисту даних. Система дозволить не тільки ефективно контролювати фізичний доступ до об'єктів, але й створить надійний захист персональних даних користувачів від різноманітних кіберзагроз.

Користувачами системи будуть як співробітники служби безпеки та IT-відділу, відповідальні за налаштування та адміністрування системи, так і всі працівники організації, які використовуватимуть її для щоденного доступу до робочих приміщень та ресурсів. Крім того, система надасть цінну інформацію для керівництва компанії щодо ефективності використання робочого часу та дотримання політик безпеки.

Таким чином, розробка та впровадження комп'ютеризованої системи контролю доступу із хешуванням персональних даних є актуальним та важливим завданням, яке дозволить організації суттєво підвищити рівень безпеки, оптимізувати процеси управління доступом та забезпечити надійний захист конфіденційної інформації в умовах зростаючих кіберзагроз та посилення вимог до захисту персональних даних.

РОЗДІЛ 2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА ОБГРУНТУВАННЯ ВИБОРУ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРИЗОВАНОЇ СИСТЕМИ ДОСТУПУ У ПРИМІЩЕННЯ

2.1 Проектування архітектури комп'ютеризованої системи контролю доступу з хешуванням персональних даних

Проектування архітектури комп'ютеризованої системи контролю доступу з хешуванням персональних даних є ключовим етапом у створенні надійної та ефективної системи безпеки. Базуючись на детальному аналізі технічного завдання, архітектура системи розроблена таким чином, щоб забезпечити високий рівень захисту персональних даних, зручність використання та гнучкість у налаштуванні. Система складається з двох основних доменів: апаратного та програмного, які тісно взаємодіють між собою для забезпечення комплексного рішення контролю доступу.

Апаратний домен системи базується на платформі Arduino UNO, яка виступає центральним елементом управління фізичними компонентами системи. До плати Arduino UNO підключений модуль MFRC522, який відповідає за зчитування та запис RFID-карток або міток. Цей модуль забезпечує швидку та надійну ідентифікацію користувачів системи. Крім того, до Arduino UNO підключений сервомотор, який виконує функцію фізичного механізму контролю доступу, наприклад, відкриття дверей або турнікету. Така конфігурація дозволяє створити повноцінну систему контролю фізичного доступу, яка може бути легко інтегрована в існуючу інфраструктуру будівлі або офісу.

Важливою особливістю апаратного домену є його з'єднання з персональним комп'ютером через COM-порт. Це з'єднання забезпечує двосторонній обмін даними між апаратною частиною та програмним забезпеченням, реалізованим на персональному комп'ютері. Таке рішення дозволяє централізовано керувати системою, оновлювати налаштування та збирати дані про активність користувачів в режимі реального часу.

Програмний домен системи реалізований на робочій станції (персональному комп'ютері) і включає в себе REST API сервер, розроблений на мові програмування Python. Вибір Python обумовлений його гнучкістю, багатою екосистемою бібліотек та простотою розробки. REST API сервер виступає центральним елементом програмної частини, забезпечуючи взаємодію між різними компонентами системи, обробку запитів від клієнтської частини та управління базою даних.

База даних системи реалізована на основі SQLite, що є легковагим рішенням для зберігання даних. Вибір SQLite обумовлений його простотою у використанні, відсутністю необхідності в окремому сервері баз даних та можливістю зберігання всієї бази даних у вигляді єдиного файлу. Це значно спрощує процеси резервного копіювання та відновлення системи у випадку збоїв або необхідності міграції на інший сервер. Додаткофою мірою захисту є можливість перенесення бази даних у вигляді єдиного файлу на флешку адміністратора та зберігання її у сейфі чи іншому надійному місці (при собі), що забезпечує надійний фізичний бар'єр для захистуданих на час простою системи. Структура бази даних складається з двох основних таблиць, що забезпечують ефективне зберігання та управління інформацією про користувачів та їх активність.

Перша таблиця бази даних призначена для зберігання інформації про працівників та їх права доступу. Вона містить наступні поля:

IDemployers [AutoNumber] унікальний ідентифікатор запису;

AccessLevel [Int Number] рівень доступу, визначає права доступу працівника до різних зон може приймати значення 1 - найнижчий рівень доступу, 2 - високий рівень доступу, 3 - найвищий рівень доступу;

Position [255 character Short Text] посада працівника

sha256Hash [64 character Short Text, PrimaryKey] хеш-значення, яке виступає як первинний ключ та забезпечує унікальну ідентифікацію користувача без зберігання його персональних даних у відкритому вигляді;

PIN [Large Number] 8 цифр персональний ідентифікаційний номер для додаткової аутентифікації.

Використання sha256Hash як первинного ключа забезпечує високий рівень захисту персональних даних, оскільки оригінальна інформація (як от прізвище чи ім'я) не зберігається в базі даних, а лише її хеш-значення.

Друга таблиця бази даних призначена для зберігання інформації про відвідуваність працівниками робочого місця. Вона містить поля:

IDpassings [AutoNumber] унікальний ідентифікатор запису;

sha256Hash [64 character Short Text, ForeignKey] який виступає як зовнішній ключ, пов'язуючи запис з інформацією про працівника з першої таблиці;

TimeStamp [YYYY.MM.DD HH:MM:SS] часова мітка, яка фіксує точний час входу або виходу працівника).

Ця таблиця дозволяє вести детальний облік активності працівників, що може бути використано для аналізу ефективності роботи, контролю дотримання робочого графіку та забезпечення безпеки приміщень.

REST API сервер, розроблений на Python, відіграє ключову роль у функціонуванні системи. Він забезпечує взаємодію між базою даних SQLite та клієнтською частиною, а також керує комунікацією з апаратною частиною через COM-порт. Сервер обробляє запити на читання та запис даних на RFID-картки або мітки, виконує операції з базою даних, включаючи додавання нових користувачів, оновлення інформації про існуючих користувачів та

генерацію звітів про активність. Важливою функцією сервера є обчислення хеш-значень sha256 для персональних даних користувачів, що забезпечує їх захист від несанкціонованого доступу.

Для реалізації хешування персональних даних REST API сервер використовує бібліотеку hashlib, яка є частиною стандартної бібліотеки Python. Ця бібліотека надає широкий спектр криптографічних хеш-функцій, включаючи SHA-256, яка використовується в даній системі. Процес хешування відбувається наступним чином: при додаванні нового користувача або оновленні даних існуючого, сервер об'єднує особисті дані (наприклад, ім'я, прізвище та по батькові) в єдиний рядок, застосовує до нього функцію SHA-256 і отримане хеш-значення зберігає в базі даних. Це забезпечує однонаправлене перетворення персональних даних, що унеможливлює їх відновлення з хеш-значення, навіть у випадку несанкціонованого доступу до бази даних.

Клієнтська частина системи реалізована з використанням веб-технологій: HTML, CSS та Vanilla JavaScript. Такий підхід забезпечує кросплатформенність рішення, дозволяючи використовувати систему на різних пристроях без необхідності встановлення додаткового програмного забезпечення. Клієнтський інтерфейс розроблений з урахуванням принципів user-friendly дизайну, що забезпечує інтуїтивно зрозуміле управління системою для користувачів з різним рівнем технічної підготовки.

Клієнтська частина має три основні режими роботи, кожен з яких призначений для виконання специфічних завдань. Перший режим забезпечує повноцінне управління базою даних SQLite. Він надає адміністраторам системи можливість виконувати всі необхідні операції з даними, включаючи створення нових записів, читання існуючої інформації, оновлення даних користувачів та видалення застарілих або непотрібних записів. Цей режим також дозволяє генерувати різноманітні звіти, наприклад, про відвідуваність

працівників за певний період часу, що може бути корисним для аналізу ефективності роботи персоналу та оптимізації робочих процесів.

Другий режим клієнтської частини представляє собою спеціалізовану форму для розрахунку та запису хеш-значень sha256. Цей режим особливо корисний при додаванні нових користувачів до системи або при необхідності оновлення даних існуючих користувачів. Адміністратор вводить необхідні персональні дані (прізвище, ім'я, по батькові) в форму, після чого система автоматично розраховує хеш-значення та зберігає його в базі даних. Крім того, цей режим дозволяє змінювати або перевіряти PIN-коди користувачів, що може бути необхідно для забезпечення додаткового рівня безпеки або у випадках, коли користувач забув свій PIN-код.

Третій режим клієнтської частини призначений для ручного управління доступом у випадках, коли апаратна частина системи з якихось причин не функціонує. Цей режим забезпечує можливість авторизації користувачів шляхом введення їх персональних даних та PIN-коду вручну. Такий підхід гарантує безперервність роботи системи контролю доступу навіть у нестандартних ситуаціях, що є критично важливим для забезпечення безпеки об'єкта.

Важливою особливістю архітектури системи є її модульність та масштабованість. Система спроектована таким чином, що дозволяє легко додавати нові функціональні можливості або розширювати існуючі без необхідності повної переробки архітектури. Наприклад, можна легко інтегрувати додаткові методи аутентифікації, такі як біометричні дані або двофакторна аутентифікація, шляхом додавання відповідних модулів до апаратної частини та розширення функціональності програмного забезпечення.

Безпека даних є одним з ключових аспектів розробленої архітектури. Крім використання хешування для захисту персональних даних, система передбачає використання шифрованого з'єднання між клієнтською частиною

та сервером, а також між сервером та базою даних. Це забезпечує захист даних під час їх передачі та зберігання. Крім того, система включає механізми логування всіх дій користувачів та адміністраторів, що дозволяє проводити аудит безпеки та виявляти потенційні загрози.

Для забезпечення надійності та відмовостійкості системи передбачені механізми автоматичного резервного копіювання бази даних. Завдяки використанню SQLite, вся база даних зберігається в єдиному файлі, що значно спрощує процес резервного копіювання. Система налаштована на регулярне створення резервних копій, які можуть бути легко відновлені у випадку збою або пошкодження основної бази даних.

Важливим аспектом архітектури є її відповідність вимогам законодавства щодо захисту персональних даних, таким як GDPR в Європейському Союзі. Система забезпечує збір та обробку тільки необхідних персональних даних, надає можливість користувачам отримувати доступ до своїх даних та видаляти їх за запитом, а також забезпечує прозорість обробки персональних даних відповідно до законодавчих вимог.

Таким чином, розроблена архітектура комп'ютеризованої системи контролю доступу з хешуванням персональних даних представляє собою комплексне рішення, яке забезпечує високий рівень безпеки, зручність використання та гнучкість налаштування. Поєднання надійної апаратної частини з ефективним програмним забезпеченням створює систему, здатну задовольнити потреби різноманітних організацій в сфері контролю доступу та захисту персональних даних співробітників.

2.2 Проектування апаратної частини комп'ютеризованої системи контролю доступу з хешуванням персональних даних

Проектування апаратної частини комп'ютеризованої системи контролю доступу з хешуванням персональних даних є ключовим етапом у створенні

надійної та ефективної системи безпеки. Основою апаратної частини виступає мікроконтролер Arduino UNO, який обрано за його оптимальне співвідношення функціональності, доступності та простоти програмування. Arduino UNO забезпечує достатню обчислювальну потужність для обробки даних RFID-міток, керування сервоприводом та взаємодії з іншими компонентами системи. Важливою перевагою Arduino UNO є наявність великої кількості цифрових та аналогових входів/виходів, що дозволяє підключити всі необхідні периферійні пристрої без використання додаткових розширювачів портів.

Для забезпечення безконтактної ідентифікації користувачів у системі використовується RFID-зчитувач MFRC522. Цей модуль працює на частоті 13,56 МГц та підтримує стандарт ISO/IEC 14443 A/MIFARE, що забезпечує сумісність з широким спектром RFID-карток та міток. MFRC522 підключається до Arduino UNO через SPI інтерфейс, використовуючи піни 10 (SDA), 11 (MOSI), 12 (MISO) та 13 (SCK). Додатково, пін RST модуля MFRC522 підключається до піну 9 Arduino для можливості програмного скидання зчитувача. Така конфігурація забезпечує швидкий та надійний обмін даними між зчитувачем та мікроконтролером.

Для реалізації фізичного механізму контролю доступу в системі використовується сервопривод MG996R. Цей сервопривод має достатній крутний момент (9,4 кг*см при напрузі 4,8В) для надійного відкриття та закриття дверей або управління турнікетом. Сигнальний провід сервоприводу підключається до піну 8 Arduino UNO, що дозволяє точно контролювати його положення. Важливо зазначити, що живлення сервоприводу здійснюється від окремого джерела живлення, а не через Arduino, щоб уникнути перевантаження мікроконтролера та забезпечити стабільну роботу всієї системи.

Взаємодія між компонентами системи відбувається наступним чином: коли користувач підносить RFID-картку до зчитувача MFRC522, модуль

зчитує унікальний ідентифікатор картки та попередньо записаний на карту sha256Hash довжиною 64 символи та передає його на Arduino UNO через SPI інтерфейс. Мікроконтролер обробляє отриману інформацію, передає її на робочу станцію (персональний компютер) через COM-port, Rest API сервером на основі Python порівнює 64 символьний хеш зі збереженими у базіданих, що зберігаються в базі SQLite базі даних системи. Якщо ідентифікатор картки відповідає авторизованому користувачу, Arduino UNO подає сигнал на сервопривод MG996R для відкриття дверей або турнікету. Одночасно з цим RGB світлодіод змінює свій колір з червоного (заборона) на зелений (дозвіл) колір, що свідчить про успішну авторизацію та надання доступу.

Для забезпечення надійної роботи системи передбачено стабільне джерело живлення. В якості основного джерела живлення використовується адаптер з вихідною напругою 5В та струмом не менше 2А. Цей адаптер забезпечує живлення Arduino UNO через вбудований стабілізатор напруги, а також живлення інших компонентів системи. Для підвищення надійності системи та забезпечення її роботи у випадку відключення електроенергії, рекомендується використовувати резервне джерело живлення у вигляді акумуляторної батареї ємністю не менше 7 А*год. Для автоматичного перемикання між основним та резервним джерелами живлення рекомендовано використовувати спеціальний модуль з функцією UPS (Uninterruptible Power Supply).

Важливим аспектом проектування апаратної частини є забезпечення захисту компонентів від зовнішніх впливів та несанкціонованого доступу. Для цього вся система розміщується в герметичному пластиковому корпусі з ступенем захисту IP65, який забезпечує захист від пилу та вологи. Всередині корпусу встановлюється монтажна плата, на якій кріпляться Arduino UNO, MFRC522, блок живлення та інші компоненти системи. Для введення кабелів у корпус використовуються спеціальні ущільнювачі, які забезпечують герметичність з'єднань.

Для підвищення загального рівня безпеки системи рекомендується додати ряд додаткових елементів. Зокрема, встановлення датчика відкриття дверей (геркона) дозволить контролювати несанкціоновані спроби проникнення. Звуковий сигналізатор може бути використаний для оповіщення про спроби несанкціонованого доступу або інші нештатні ситуації. Крім того, важливо передбачити резервну кнопку аварійного відкриття дверей, яка дозволить забезпечити вихід людей з приміщення у випадку відмови основної системи контролю доступу або надзвичайної ситуації.

Розроблена апаратна частина системи передбачає можливість масштабування та додавання нових функцій. Наприклад, для контролю декількох точок доступу можна використовувати додаткові RFID-зчитувачі, підключені через мультиплексор. Це дозволить розширити зону контролю системи без необхідності встановлення додаткових мікроконтролерів. Також існує можливість інтеграції додаткових біометричних сенсорів, таких як сканери відбитків пальців або модулі розпізнавання обличчя. Ці компоненти можуть бути підключені до вільних портів Arduino UNO або через додатковий мікроконтролер, з'єднаний з основним через інтерфейс I2C або UART.

Для забезпечення можливості віддаленого доступу та моніторингу системи може бути додано модуль Wi-Fi (наприклад, ESP8266) або Ethernet-модуль. Це дозволить інтегрувати систему контролю доступу в загальну мережеву інфраструктуру об'єкта та забезпечити централізоване управління та моніторинг. При цьому важливо забезпечити належний рівень захисту мережевого з'єднання, використовуючи сучасні протоколи шифрування та аутентифікації.

Особлива увага при проектуванні апаратної частини приділяється забезпеченню стабільності та надійності роботи системи. Для цього всі з'єднання між компонентами виконуються з використанням якісних роз'ємів та кабелів, що мінімізує ризик виникнення проблем через погані контакти.

Крім того, для захисту від електромагнітних завад використовуються екрановані кабелі та додаткові фільтруючі елементи в ланцюгах живлення.

Важливим етапом розробки апаратної частини є тестування та налагодження системи. Проводиться ряд тестів для перевірки стабільності живлення всіх компонентів, надійності зчитування RFID-міток на різних відстанях, точності роботи системи в різних умовах експлуатації. Особлива увага приділяється тестуванню механізму відкриття/закриття дверей на надійність та швидкодію, а також перевірці системи на стійкість до електромагнітних завад. Всі виявлені під час тестування недоліки усуваються шляхом доопрацювання конструкції або корекції програмного забезпечення.

Таким чином, спроектована апаратна частина комп'ютеризованої системи контролю доступу з хешуванням персональних даних забезпечує надійну основу для реалізації всіх необхідних функцій системи. Вона поєднує в собі високу функціональність, надійність та гнучкість, що дозволяє адаптувати систему до різних умов експлуатації та вимог безпеки. Використання сучасних компонентів та технологій, а також продумана архітектура системи гарантують її ефективну роботу та можливість подальшого розширення функціоналу відповідно до зростаючих потреб користувачів та вимог безпеки.

2.3 Обгрунтування вибору мікроконтролера для управління процесом зчитування та передачі даних

Обґрунтування вибору мікроконтролера для управління процесом зчитування та передачі даних є критично важливим етапом у проектуванні комп'ютеризованої системи контролю доступу. Від цього вибору залежить не лише функціональність та ефективність системи, але й її надійність, масштабованість та економічна доцільність. Після ретельного аналізу доступних на ринку рішень та зважування всіх переваг та недоліків різних

платформ, було прийнято рішення використати плату Arduino UNO (рис. 2.1), яка базується на мікроконтролері ATmega328P [12]. Це рішення оптимально відповідає всім вимогам проекту, забезпечуючи необхідну продуктивність, гнучкість та простоту розробки.

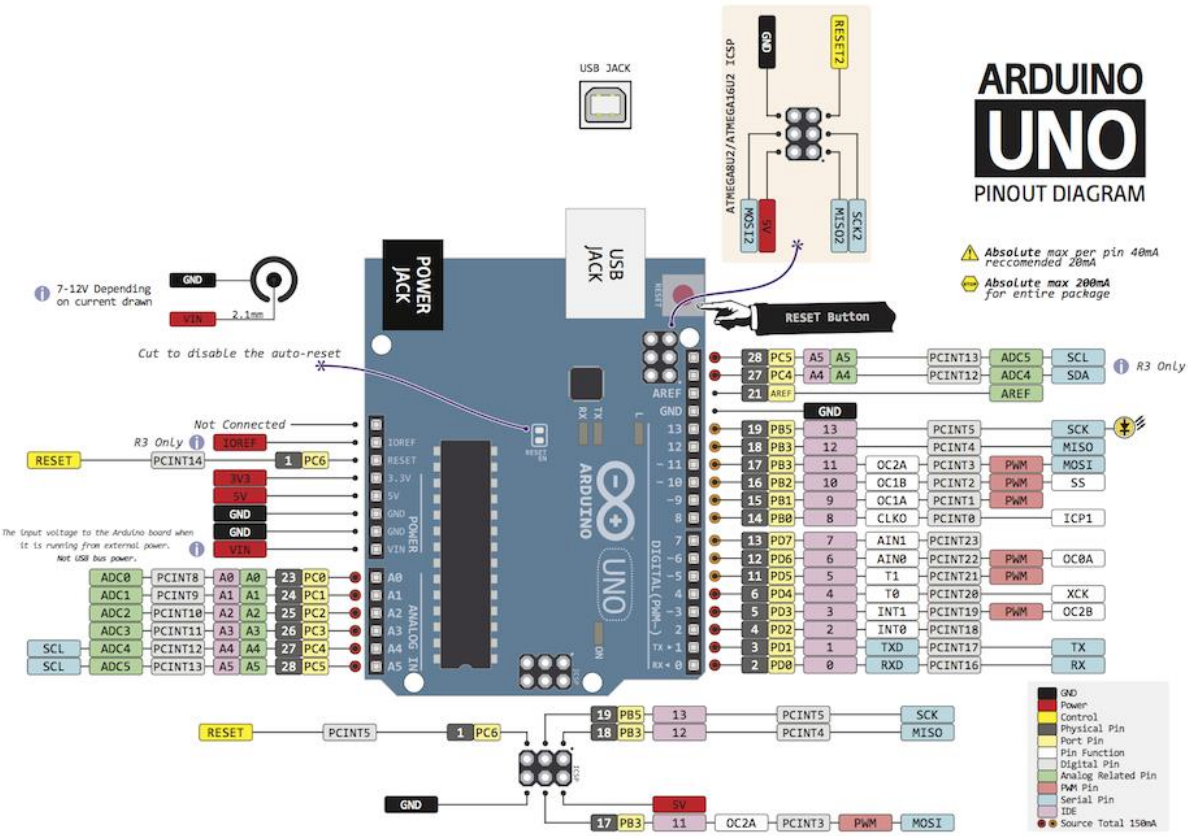

Рисунок 2.1 – Плата Arduino UNO

Arduino UNO представляє собою популярну платформу для розробки, яка завоювала визнання як професійних інженерів, так і ентузіастів електроніки по всьому світу. В основі цієї плати лежить потужний для ембедед-систем 8-бітний мікроконтролер ATmega328P, який працює на тактовій частоті 16 МГц. Така частота забезпечує достатню обчислювальну потужність для швидкої та ефективної обробки даних, отриманих з RFID-міток, а також для керування всіма периферійними пристроями системи контролю доступу. Arduino UNO оснащена 32 КБ флеш-пам'яті, яка використовується для зберігання програмного коду. Цього об'єму цілком

достатньо для реалізації всіх необхідних алгоритмів обробки RFID-даних, логіки керування доступом, а також для зберігання додаткових бібліотек, які можуть знадобитися в процесі розробки. Крім того, плата має 2 КБ оперативної пам'яті SRAM, яка використовується для зберігання змінних під час виконання програми, та 1 КБ енергонезалежної пам'яті EEPROM, яка може бути використана для довготривалого зберігання важливих даних, таких як налаштування системи або журнал подій.

Одним з ключових переваг Arduino UNO є наявність 14 цифрових входів/виходів, 6 з яких можуть бути використані як ШІМ-виходи. Це дозволяє підключити всі необхідні компоненти системи контролю доступу, включаючи RFID-зчитувач, сервопривід для керування механізмом відкриття дверей, світлодіоди для індикації стану системи, кнопки для ручного керування та інші елементи. Крім того, наявність 6 аналогових входів розширює можливості системи, дозволяючи підключати різноманітні сенсори, наприклад, датчики температури або освітленості, які можуть бути використані для додаткового моніторингу умов навколишнього середовища. Важливою особливістю Arduino UNO є підтримка різних комунікаційних інтерфейсів, таких як UART, I2C та SPI. Це забезпечує гнучкість при виборі периферійних пристроїв та дозволяє легко інтегрувати систему контролю доступу з іншими системами безпеки або автоматизації будівлі.

Мікроконтролер ATmega328P, який є серцем Arduino UNO, заслуговує окремої уваги. Цей 8-бітний мікроконтролер архітектури AVR має ряд характеристик, які роблять його ідеальним вибором для системи контролю доступу. Його архітектура RISC з 131 потужними інструкціями забезпечує високу ефективність виконання програмного коду, що особливо важливо для систем майже реального часу, якими є системи контролю доступу. ATmega328P здатний досягати продуктивності до 20 MIPS при тактовій частоті 20 МГц, що з запасом покриває потреби нашої системи. Важливою характеристикою цього мікроконтролера є його низьке енергоспоживання: в

активному режимі при частоті 1 МГц та напрузі живлення 1.8В він споживає лише 0.2 мА, а в режимі енергозбереження споживання може знижуватися до 0.1 мкА. Це особливо важливо для систем, які можуть працювати від резервного живлення, наприклад, при відключенні основного електропостачання.

ATmega328P (рис. 2.2) має багату периферію, яка включає в себе два 8-бітних та один 16-бітний таймери/лічильники, годинник реального часу з окремим осцилятором, 6 каналів ШІМ для точного керування сервоприводами або світлодіодами, 6-канальний 10-бітний АЦП для роботи з аналоговими сенсорами, програмований послідовний USART для комунікації з іншими пристроями, а також інтерфейси SPI та I2C для підключення різноманітних периферійних модулів.

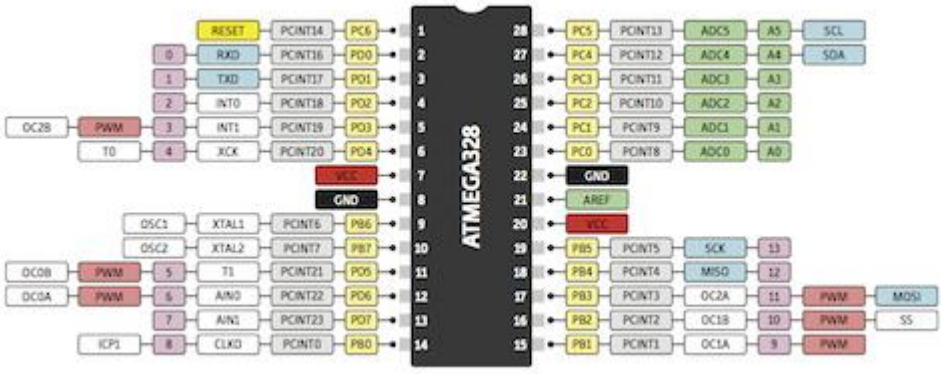

Рисунок 2.2 – Мікроконтролер ATmega328P

Особливо варто відзначити наявність програмованого сторожового таймера з окремим вбудованим осцилятором, який підвищує надійність системи, запобігаючи зависанням програми. Це критично важливо для систем безпеки, де будь-який збій може призвести до серйозних наслідків.

Широкий діапазон робочих напруг ATmega328P (від 1.8В до 5.5В) та температурний діапазон від -40°C до 85°C роблять його придатним для використання в різних умовах експлуатації, що особливо важливо для систем

контролю доступу, які можуть бути встановлені як всередині приміщень, так і на вулиці. Наявність внутрішнього каліброваного RC-осцилятора підвищує точність роботи мікроконтролера та зменшує залежність від зовнішніх компонентів, що спрощує проектування друкованої плати та підвищує надійність системи в цілому.

Вибір Arduino UNO на базі ATmega328P для системи контролю доступу обґрунтований рядом факторів. По-перше, ця платформа забезпечує оптимальне співвідношення продуктивності та енергоспоживання. Тактова частота 16 МГц дозволяє швидко обробляти дані з RFID-міток, керувати сервоприводом без помітних затримок та одночасно виконувати інші завдання, такі як моніторинг стану датчиків або комунікація з іншими пристроями. При цьому енергоспоживання залишається на прийнятному рівні, що важливо для систем, які повинні працювати безперервно.

По-друге, обсяг пам'яті Arduino UNO оптимально відповідає вимогам проекту. 32 КБ флеш-пам'яті достатньо для зберігання всього необхідного програмного коду, включаючи бібліотеки для роботи з RFID-зчитувачем, алгоритми обробки даних та логіку керування доступом. 2 КБ SRAM забезпечують достатній простір для зберігання змінних та проміжних даних під час виконання програми, а 1 КБ EEPROM може бути використаний для зберігання налаштувань системи або журналу подій, які повинні зберігатися навіть при відключенні живлення.

Третім важливим фактором є різноманітність доступних інтерфейсів. Наявність UART, SPI та I2C дозволяє легко інтегрувати в систему різні периферійні пристрої. Наприклад, RFID-зчитувач MFRC522 може бути підключений через інтерфейс SPI, що забезпечує швидкий та надійний обмін даними. I2C інтерфейс може бути використаний для підключення додаткових модулів, таких як годинник реального часу або датчик температури та вологості. UART інтерфейс може бути використаний для комунікації з комп'ютером або іншими системами автоматизації будівлі.

Четвертим аргументом на користь вибору Arduino UNO є її надійність та стійкість до електромагнітних завад. ATmega328P має вбудовані механізми захисту від збоїв, такі як сторожовий таймер та детектор падіння напруги живлення. Це особливо важливо для систем безпеки, де надійність є критичним фактором. Крім того, плата Arduino UNO має захист від перевантаження по струму та неправильної полярності підключення, що підвищує її стійкість до помилок при монтажі та експлуатації.

П'ятим важливим фактором є простота програмування та наявність великої спільноти розробників. Arduino IDE надає зручне середовище для розробки та налагодження програм, а багата екосистема бібліотек значно спрощує реалізацію різних функцій системи. Наприклад, для роботи з RFID-зчитувачем MFRC522 існують готові бібліотеки, які дозволяють швидко інтегрувати цей компонент в систему. Велика спільнота розробників Arduino забезпечує доступ до великої кількості прикладів коду, навчальних матеріалів та форумів підтримки, що прискорює процес розробки та дозволяє ефективно вирішувати можливі проблеми.

Шостим аргументом є достатня кількість входів/виходів. 14 цифрових та 6 аналогових портів Arduino UNO дозволяють підключити всі необхідні компоненти системи контролю доступу без необхідності використання додаткових розширювачів портів. Це спрощує схему системи, підвищує її надійність та знижує вартість. Наприклад, можна одночасно підключити RFID-зчитувач через SPI інтерфейс, сервопривід для керування замком, світлодіоди для індикації стану системи, кнопки для ручного керування та датчик відкриття дверей.

Сьомою перевагою Arduino UNO є можливість легкого розширення системи. При необхідності додавання нових функцій або збільшення кількості контрольованих точок доступу, система може бути легко розширена за рахунок додаткових модулів або переходу на потужніші плати Arduino, такі як

Arduino Mega, без значних змін у програмному коді. Це забезпечує гнучкість системи та можливість її адаптації до змінних вимог замовника.

Восьмим фактором є доступність та економічна ефективність платформи Arduino UNO. Ця плата широко доступна на ринку, має помірну вартість та не вимагає дорогих інструментів для програмування та налагодження. Це важливо для оптимізації витрат на розробку та виробництво системи контролю доступу, особливо при невеликих обсягах виробництва або при розробці прототипів.

Дев'ятою перевагою є температурна стійкість мікроконтролера ATmega328P. Широкий робочий діапазон температур від -40°C до 85°C дозволяє використовувати систему контролю доступу в різних умовах експлуатації, включаючи зовнішні установки в регіонах з екстремальним кліматом. Це розширює сферу застосування системи та підвищує її конкурентоспроможність на ринку.

Десятим аргументом на користь вибору Arduino UNO є вбудовані функції безпеки мікроконтролера ATmega328P. Наявність сторожового таймера дозволяє автоматично перезавантажувати систему у випадку зависання програми, що критично важливо для забезпечення безперервної роботи системи контролю доступу. Крім того, можливість програмного блокування секторів пам'яті запобігає несанкціонованому доступу до критичних даних та налаштувань системи.

У контексті побудови системи контролю доступу, Arduino UNO з мікроконтролером ATmega328P забезпечує всі необхідні функції та характеристики. Вона дозволяє реалізувати швидке та надійне зчитування даних з RFID-міток, забезпечує стабільне керування механізмом відкриття та закриття дверей за допомогою сервоприводу, підтримує підключення додаткових сенсорів для підвищення рівня безпеки, наприклад, датчиків відкриття дверей або руху. Система може бути легко доповнена звуковою та світловою індикацією для інформування користувачів про стан доступу.

Завдяки низькому енергоспоживанню та можливості роботи від широкого діапазону напруг живлення, система може стабільно функціонувати в різних умовах експлуатації, включаючи роботу від резервних джерел живлення.

Крім того, Arduino UNO забезпечує можливість легкої інтеграції системи контролю доступу з іншими системами безпеки та автоматизації будівлі. Наприклад, через UART інтерфейс система може передавати дані про події доступу на центральний сервер для подальшого аналізу та генерації звітів. Можливість програмування на мові C++ та наявність великої кількості готових бібліотек дозволяє реалізувати складні алгоритми обробки даних та логіки контролю доступу, включаючи можливість реалізації базових криптографічних функцій для підвищення безпеки системи.

Варто також відзначити, що вибір Arduino UNO не обмежує можливості подальшого розвитку та вдосконалення системи. При необхідності розширення функціоналу або збільшення кількості контрольованих точок доступу, система може бути легко модифікована шляхом додавання нових модулів або переходу на більш потужні платформи Arduino, такі як Arduino Mega або Arduino Due, без значних змін у програмному коді. Це забезпечує гнучкість рішення та можливість його адаптації до зростаючих потреб користувачів.

Однак, варто зазначити і деякі обмеження Arduino UNO, які слід враховувати при проектуванні системи контролю доступу:

1. Обмежений обсяг пам'яті: 32 КБ флеш-пам'яті та 2 КБ SRAM можуть бути недостатніми для реалізації дуже складних алгоритмів обробки даних або зберігання великої кількості записів про користувачів системи. У таких випадках може знадобитися використання зовнішньої пам'яті або перехід на більш потужну платформу.

2. Відсутність вбудованого Ethernet або Wi-Fi модуля: для реалізації мережевих функцій потрібно використовувати додаткові модулі, що може збільшити вартість системи та ускладнити її конструкцію.

3. Обмежена обчислювальна потужність: хоча для більшості завдань контролю доступу потужності ATmega328P цілком достатньо, для реалізації складних криптографічних алгоритмів або обробки великих обсягів даних в реальному часі може знадобитися більш потужний процесор.

4. Відсутність апаратного прискорення криптографічних операцій: це може обмежити можливості реалізації складних схем шифрування безпосередньо на мікроконтролері.

5. Обмежені можливості багатозадачності: Arduino UNO не має операційної системи реального часу, що може ускладнити реалізацію складних сценаріїв роботи з одночасним виконанням багатьох завдань.

Незважаючи на ці обмеження, для більшості типових сценаріїв використання систем контролю доступу, Arduino UNO залишається оптимальним вибором, забезпечуючи необхідну функціональність, надійність та економічну ефективність.

У контексті розробки комп'ютеризованої системи контролю доступу з хешуванням персональних даних, Arduino UNO може ефективно виконувати роль інтерфейсу між апаратною частиною (RFID-зчитувач, сервопривід, датчики) та програмною частиною системи, реалізованою на персональному комп'ютері. Мікроконтролер може здійснювати зчитування даних з RFID-міток, попередню обробку цих даних, керування механізмом відкриття/закриття дверей та передачу інформації на ПК для подальшої обробки та зберігання.

При цьому, більш складні операції, такі як хешування персональних даних, перевірка прав доступу та ведення бази даних користувачів, можуть бути реалізовані на стороні ПК, де доступні більші обчислювальні ресурси та спеціалізовані бібліотеки для роботи з криптографічними функціями. Така архітектура дозволяє оптимально розподілити навантаження між компонентами системи, забезпечуючи високу продуктивність та надійність при збереженні економічної ефективності рішення.

Таким чином, вибір Arduino UNO на базі мікроконтролера ATmega328P для управління процесом зчитування та передачі даних в комп'ютеризованій системі контролю доступу з хешуванням персональних даних є обґрунтованим та оптимальним рішенням. Ця платформа забезпечує необхідну функціональність, надійність та гнучкість, дозволяючи створити ефективну систему контролю доступу, яка відповідає сучасним вимогам безпеки та може бути легко адаптована до різних умов експлуатації та зростаючих потреб користувачів.

2.4 Обґрунтування вибору модуля MFRC522 для зчитування та запису RFID карт і міток

Обґрунтування вибору модуля MFRC522 [13] для зчитування та запису RFID карт і міток у розробленій комп'ютеризованій системі контролю доступу з хешуванням персональних даних базується на комплексному аналізі технічних, економічних та практичних факторів. MFRC522, розроблений компанією NXP Semiconductors, є високоінтегрованим модулем для безконтактного зв'язку на частоті 13,56 МГц, що повністю відповідає стандарту ISO/IEC 14443A. Ця відповідність стандарту забезпечує широку сумісність з різноманітними RFID-картками та мітками, які широко використовуються в сучасних системах контролю доступу.

Модуль підтримує різні протоколи, включаючи MIFARE Classic, MIFARE Ultralight, MIFARE DESFire, MIFARE Plus та інші ISO 14443A-сумісні протоколи, що надає системі необхідну гнучкість та можливість адаптації до різних типів ідентифікаторів. Ця універсальність є особливо важливою в контексті розробки системи контролю доступу, оскільки дозволяє організаціям використовувати вже наявні RFID-картки або легко інтегрувати нові типи ідентифікаторів без необхідності заміни всього обладнання.

Відстань зчитування MFRC522, яка досягає 50 мм для стандартних RFID-карток, є оптимальною для систем контролю доступу, забезпечуючи зручність використання без компромісу щодо безпеки. Ця характеристика дозволяє користувачам швидко та ефективно проходити точки контролю, просто піднісши картку до зчитувача, що особливо важливо в умовах високої прохідності. Швидкість передачі даних до 848 кбіт/с в режимі зчитування забезпечує миттєву обробку інформації та мінімальні затримки при авторизації користувачів, що є критичним фактором для забезпечення комфорту та ефективності роботи системи в цілому. Ця висока швидкість також дозволяє системі обробляти велику кількість запитів на доступ у короткий проміжок часу, що особливо важливо для великих організацій з інтенсивним потоком співробітників та відвідувачів.

Низьке енергоспоживання MFRC522, яке складає від 13 до 26 мА в активному режимі та менше 80 мкА в режимі очікування, є важливим фактором для загальної енергоефективності системи. Це особливо актуально в контексті сучасних тенденцій до створення екологічно чистих та енергоефективних рішень. Низьке енергоспоживання також зменшує навантаження на джерела живлення системи, що підвищує її надійність та зменшує витрати на експлуатацію.

Підтримка SPI інтерфейсу з максимальною швидкістю до 10 Мбіт/с забезпечує швидкий та надійний зв'язок з мікроконтролером Arduino UNO, що є ключовим елементом розробленої системи. Ця висока швидкість інтерфейсу дозволяє ефективно передавати не лише ідентифікаційні дані, але й додаткову інформацію, яка може бути корисною для розширених функцій системи контролю доступу, таких як облік робочого часу або моніторинг переміщень персоналу.

Вбудовані функції безпеки MFRC522, включаючи механізми захисту від колізій та виявлення помилок, підвищують надійність роботи системи в умовах можливих електромагнітних перешкод та спроб несанкціонованого

доступу. Ці функції особливо важливі в промислових умовах або в середовищах з високим рівнем електромагнітних завад, де стабільність роботи системи контролю доступу є критично важливою для забезпечення безпеки об'єкта.

Компактні розміри модуля (40х60 мм) та наявність монтажних отворів значно спрощують процес інтеграції MFRC522 в корпус системи контролю доступу, що дозволяє створювати естетичні та ергономічні рішення, які легко вписуються в різноманітні інтер'єри. Ця компактність також дозволяє розміщувати зчитувачі в обмежених просторах, таких як вузькі дверні рами або турнікети, не порушуючи загальний дизайн приміщення.

Напруга живлення 3,3В є сумісною з логічними рівнями Arduino UNO (що має вихід живлення 3V3) та інших компонентів системи, що спрощує проектування електричної схеми та зменшує ризик пошкодження компонентів через невідповідність напруг. Ця уніфікація напруги живлення також зменшує складність системи, оскільки не вимагає додаткових перетворювачів рівнів сигналів, що в свою чергу підвищує надійність та зменшує вартість виробництва.

Широкий температурний діапазон роботи від -20°C до +80°C дозволяє використовувати MFRC522 як в приміщеннях, так і на вулиці за умови відповідного захисту від вологи. Ця характеристика розширює сферу застосування системи, дозволяючи встановлювати точки контролю доступу не лише всередині будівель, але й на зовнішніх периметрах об'єктів, що особливо важливо для забезпечення комплексної безпеки територій.

Наявність великої кількості готових бібліотек та прикладів коду для роботи з Arduino значно спрощує процес розробки та інтеграції модуля в систему. Зокрема, бібліотека MFRC522 для Arduino надає зручні функції для ініціалізації модуля, зчитування та запису даних на RFID-картки, що дозволяє швидко реалізувати необхідний функціонал та зосередитися на розробці

специфічних для проекту функцій, таких як підшотовка до хешування персональних даних та подальша інтеграція з базою даних.

Ця багата екосистема, яка включає карти, мітки, стікери табрелки, як зображено на рисунку 2.3, також забезпечує можливість швидкого вирішення потенційних проблем та оптимізації роботи системи на основі досвіду інших розробників. З точки зору безпеки, підтримка технології MIFARE Classic з використанням криптографічного алгоритму Crypto1 для захисту даних є додатковим шаром захисту. Хоча цей алгоритм має відомі вразливості, в контексті розробленої системи з хешуванням персональних даних це не є критичним, оскільки на картках зберігаються лише хеш-значення, а не самі персональні дані. Це дозволяє створити багаторівневу систему захисту, де навіть у випадку компрометації RFID-картки, зловмисник не отримає доступу до реальних персональних даних користувачів.

Рисунок 2.3 – Мікроконтролер ATmega328P

Економічні фактори також відіграють важливу роль у виборі MFRC522. Модуль має відносно низьку вартість при високій функціональності, що робить його оптимальним вибором для систем середнього масштабу. Це дозволяє організаціям впроваджувати надійні системи контролю доступу без надмірних витрат на обладнання. Крім того, широка доступність модуля на ринку забезпечує можливість швидкої заміни у випадку виходу з ладу та

спрощує процес обслуговування системи, що є важливим фактором для забезпечення безперебійної роботи системи контролю доступу в довгостроковій перспективі.

При виборі MFRC522 були враховані і його обмеження. Модуль не підтримує роботу з NFC-пристроями в режимі емуляції карти, що може обмежити можливості використання смартфонів як ідентифікаторів. Однак, для базової системи контролю доступу це не є критичним обмеженням, оскільки більшість організацій все ще віддають перевагу фізичним RFID-карткам через їх надійність та простоту управління та можливість вилучення непотрібних чи скомпрометованих ключів.

MFRC522 також не має вбудованого антиколізійного механізму для роботи з кількома картками одночасно, але в контексті системи контролю доступу, де зазвичай зчитується одна картка за раз, це обмеження не є суттєвим.

Відсутність вбудованого шифрування для захисту каналу зв'язку між модулем та мікроконтролером враховано в архітектурі системи, і безпека забезпечується на рівні обробки даних в мікроконтролері та сервері та зумовлює відповідні конструктивні особливості корпуса системи. Ці обмеження були ретельно проаналізовані та визнані прийнятними в контексті розробленої системи, де переваги модуля значно перевищують його недоліки.

В процесі вибору модуля для системи контролю доступу були розглянуті й альтернативні варіанти, такі як PN532 та RC522. PN532 має ширші можливості, включаючи підтримку NFC, що потенційно дозволило б використовувати смартфони як ідентифікатори. Однак, цей модуль має вищу вартість та складнішу інтеграцію, що могло б збільшити загальну вартість системи та ускладнити процес розробки. RC522, хоча і є дуже схожим за характеристиками до MFRC522, має меншу кількість готових бібліотек та прикладів коду, що могло б уповільнити процес розробки та інтеграції. Після ретельного аналізу всіх факторів, MFRC522 був визнаний оптимальним

вибором для розробленої системи контролю доступу з хешуванням персональних даних. Цей модуль забезпечує необхідну функціональність, надійність та безпеку при оптимальному співвідношенні ціни та якості. Його широка підтримка спільнотою розробників, сумісність з різними типами RFID-карток та простота інтеграції з Arduino UNO роблять його ідеальним компонентом для створення ефективної та надійної системи контролю доступу. Використання MFRC522 дозволяє реалізувати всі необхідні функції системи, включаючи швидку ідентифікацію користувачів, безпечне зберігання та обробку даних, а також забезпечити можливість подальшого розширення та вдосконалення системи відповідно до зростаючих потреб організації в сфері безпеки та контролю доступу. Таким чином, вибір модуля MFRC522 є обґрунтованим та оптимальним рішенням для розробленої комп'ютеризованої системи контролю доступу з хешуванням персональних даних, яке забезпечує надійну основу для створення ефективної, безпечної та масштабованої системи.

2.5 Обґрунтування вибору елемента індикації

Обґрунтування вибору елемента індикації є критично важливим аспектом у процесі проектування комп'ютеризованої системи контролю доступу, оскільки цей компонент відіграє ключову роль у забезпеченні візуального зворотного зв'язку з користувачами, інформуючи їх про поточний стан системи та результати процесу ідентифікації. У рамках даного проекту, після ретельного аналізу різноманітних варіантів, в якості основного елемента індикації було обрано RGB світлодіод. Це рішення базується на цілому ряді суттєвих переваг та унікальних характеристик даного компонента, які роблять його оптимальним вибором для застосування в системах контролю доступу.

RGB світлодіод являє собою високотехнологічний електронний пристрій, який інтегрує в єдиному компактному корпусі три окремих

світлодіоди - червоний (Red), зелений (Green) та синій (Blue). Така інноваційна конструкція надає можливість створювати надзвичайно широкий спектр кольорів та відтінків шляхом варіювання інтенсивності світіння кожного з трьох базових кольорових компонентів. Ця унікальна властивість RGB світлодіода відкриває перед розробниками системи контролю доступу практично необмежені можливості для реалізації гнучкої та інформативної системи візуальної комунікації, здатної ефективно відображати різноманітні стани системи та передавати користувачам широкий спектр повідомлень, використовуючи при цьому лише один фізичний компонент.

Однією з ключових переваг використання RGB світлодіода в системі контролю доступу є можливість створення інтуїтивно зрозумілої та легко інтерпретованої користувачами системи кольорового кодування для відображення результатів процесу ідентифікації та авторизації. Наприклад, яскравий зелений колір може слугувати універсальним сигналом про успішне проходження процедури авторизації та надання доступу до захищеної зони або ресурсу. Червоний колір, навпаки, може використовуватися для індикації відмови в доступі, що може бути спричинено різними факторами, такими як недостатній рівень прав доступу, помилка в процесі ідентифікації або спроба несанкціонованого проникнення. Синій колір може бути задіяний для сигналізації про те, що система знаходиться в режимі очікування або здійснює обробку отриманих даних. Крім того, використання комбінацій різних кольорів, їх послідовного або синхронного блимання з різною частотою та інтенсивністю надає можливість створювати більш складні та інформативні візуальні повідомлення. Наприклад, чергування червоного та синього кольорів може сигналізувати про виникнення нештатної ситуації, що вимагає негайного втручання технічного персоналу, а повільне пульсування жовтого кольору може вказувати на низький рівень заряду резервної батареї системи, сигналізуючи про необхідність проведення профілактичного обслуговування.

Енергоефективність є ще одним критично важливим фактором, який суттєво вплинув на вибір RGB світлодіода в якості елемента індикації для системи контролю доступу. Сучасні світлодіодні технології характеризуються надзвичайно низьким рівнем енергоспоживання при збереженні високої яскравості та інтенсивності світіння. Ця властивість набуває особливого значення в контексті систем, які повинні зберігати працездатність в умовах автономної роботи або при живленні від резервних джерел енергії. Використання енергоефективних RGB світлодіодів дозволяє суттєво знизити загальне енергоспоживання системи контролю доступу, що призводить до зменшення навантаження на блок живлення та збільшення часу автономної роботи системи у випадках аварійного відключення основного електропостачання. Це, в свою чергу, підвищує надійність та безперебійність функціонування всієї системи контролю доступу, що є критично важливим фактором для забезпечення безпеки об'єктів з високим рівнем захисту.

Довговічність RGB світлодіодів є ще однією вагомою перевагою, яка суттєво вплинула на вибір цього компонента для системи індикації. На відміну від традиційних джерел світла, таких як лампи розжарювання або люмінесцентні індикатори, сучасні світлодіоди характеризуються надзвичайно тривалим терміном експлуатації, який може досягати десятків і навіть сотень тисяч годин безперервної роботи. Ця властивість має особливе значення для систем контролю доступу, які повинні функціонувати в режимі 24/7 без перерв на технічне обслуговування або заміну компонентів. Використання довговічних RGB світлодіодів дозволяє суттєво знизити частоту та обсяг необхідних профілактичних робіт, мінімізувати ризики виходу з ладу елементів індикації та, як наслідок, підвищити загальну надійність та безвідмовність роботи всієї системи контролю доступу. Крім того, тривалий термін служби світлодіодів сприяє зниженню загальної вартості володіння системою за рахунок зменшення витрат на технічне

обслуговування та заміну компонентів протягом всього життєвого циклу обладнання.

Компактні габаритні розміри RGB світлодіодів є ще одним важливим фактором, який обумовлює їх вибір для використання в системах контролю доступу. Мініатюрність цих компонентів дозволяє легко інтегрувати їх практично в будь-яку конструкцію системи без необхідності внесення суттєвих змін в дизайн корпусу або використання додаткових монтажних елементів. Ця властивість набуває особливого значення при встановленні системи контролю доступу в місцях з обмеженим простором, таких як вузькі дверні рами, турнікети або інші елементи інфраструктури будівель. Компактність RGB світлодіодів також сприяє створенню естетично привабливого та ергономічного дизайну системи контролю доступу, що є важливим фактором для об'єктів, де зовнішній вигляд обладнання має значення (наприклад, в офісних центрах преміум-класу або музеях). Крім того, невеликі розміри світлодіодів дозволяють при необхідності використовувати кілька елементів індикації в рамках однієї системи, що розширює можливості для створення більш складних та інформативних схем візуальної комунікації з користувачами.

З точки зору програмного управління, RGB світлодіоди надають розробникам систем контролю доступу надзвичайно широкі можливості для реалізації різноманітних сценаріїв індикації та візуальних ефектів. Використання технології широтно-імпульсної модуляції (ШІМ) для керування яскравістю кожного з трьох базових кольорових компонентів дозволяє створювати плавні та естетично привабливі переходи між різними кольорами, реалізовувати ефекти пульсації або блимання з різною частотою та інтенсивністю, а також генерувати складні візуальні патерни для передачі більш детальної інформації про стан системи. Ця гнучкість в управлінні RGB світлодіодом надає можливість створювати більш інформативні та інтуїтивно зрозумілі для користувачів системи індикації, що підвищує загальну

ергономіку та зручність експлуатації системи контролю доступу. Крім того, програмована природа RGB світлодіодів дозволяє легко адаптувати систему індикації до специфічних вимог конкретного об'єкта або змінювати схеми візуальної комунікації в процесі експлуатації без необхідності фізичної заміни компонентів.

Важливим аспектом при виборі RGB світлодіода в якості елемента індикації для системи контролю доступу є його повна сумісність з мікроконтролером Arduino UNO, який використовується в якості центрального керуючого елемента системи. Arduino UNO має вбудовані апаратні та програмні засоби для роботи з широтно-імпульсною модуляцією, що значно спрощує процес програмування та управління RGB світлодіодом. Ця сумісність дозволяє розробникам легко інтегрувати функції управління індикацією в загальну логіку роботи системи контролю доступу, реалізовувати складні алгоритми візуалізації та швидко адаптувати систему до нових вимог або сценаріїв використання. Крім того, низька напруга живлення RGB світлодіодів (зазвичай 3.3В або 5В) ідеально відповідає робочим характеристикам Arduino UNO, що дозволяє підключати світлодіоди безпосередньо до цифрових або ШІМ-виходів мікроконтролера без необхідності використання додаткових схем узгодження рівнів напруги або підсилювальних каскадів. Це не тільки спрощує загальну архітектуру системи, але й підвищує її надійність за рахунок мінімізації кількості додаткових компонентів.

При виборі конкретної моделі RGB світлодіода для використання в системі контролю доступу необхідно враховувати цілий ряд важливих технічних параметрів, які безпосередньо впливають на ефективність та зручність експлуатації системи індикації. Одним з ключових параметрів є яскравість світіння, яка повинна бути достатньою для забезпечення чіткої видимості індикації в різноманітних умовах освітлення, включаючи яскраве сонячне світло або інтенсивне штучне освітлення в приміщеннях. Кут

розсіювання світла також є важливим фактором, який впливає на видимість індикації з різних кутів огляду. Для систем контролю доступу оптимальним вибором часто є світлодіоди з достатньо широким кутом розсіювання (наприклад, 120 або 140 градусів), що забезпечує хорошу видимість індикації для користувачів, які підходять до точки контролю доступу з різних напрямків. Споживаний струм є ще одним важливим параметром, який необхідно враховувати для забезпечення оптимального балансу між яскравістю індикації та загальним енергоспоживанням системи. Тип корпусу світлодіода також має значення, особливо в контексті його інтеграції в конструкцію системи контролю доступу та забезпечення захисту від зовнішніх впливів, таких як волога або пил.

У контексті забезпечення безпеки використання RGB світлодіода в якості елемента індикації для системи контролю доступу має додаткову, але надзвичайно важливу перевагу. На відміну від текстових дисплеїв або інших типів індикаторів, які можуть відображати конкретну інформацію про користувача або його рівень доступу, кольорова індикація забезпечує високий рівень конфіденційності, не розкриваючи жодних специфічних деталей про особу, яка проходить процедуру ідентифікації. Це суттєво підвищує загальний рівень безпеки системи, мінімізуючи ризики витоку конфіденційної інформації через візуальний канал. Наприклад, сторонній спостерігач, навіть маючи можливість бачити індикацію системи, не зможе визначити, який саме рівень доступу має конкретний користувач або до яких саме ресурсів він отримує доступ. Ця властивість RGB світлодіодів робить їх ідеальним вибором для систем контролю доступу, які експлуатуються в умовах підвищених вимог до конфіденційності та захисту персональних даних.

Ще одним важливим аспектом використання RGB світлодіодів в системах контролю доступу є їх висока стійкість до різноманітних зовнішніх впливів, включаючи механічні навантаження, вібрації, перепади температур та вологості. Ця властивість набуває особливого значення для систем, які

експлуатуються в складних умовах навколишнього середовища, наприклад, на промислових об'єктах, в зонах з екстремальним кліматом або в місцях з високим рівнем електромагнітних завад. Стійкість RGB світлодіодів до зовнішніх факторів забезпечує стабільність роботи системи індикації в широкому діапазоні умов експлуатації, що підвищує загальну надійність та довговічність всієї системи контролю доступу.

Крім того, висока стійкість RGB світлодіодів до зовнішніх впливів дозволяє мінімізувати витрати на технічне обслуговування та ремонт системи індикації протягом всього життєвого циклу обладнання. Це особливо важливо для об'єктів з обмеженим доступом або віддалених локацій, де проведення регулярного технічного обслуговування може бути ускладненим або економічно недоцільним.

Важливо також відзначити, що використання RGB світлодіодів в системах контролю доступу надає можливість реалізації додаткових функцій, які виходять за рамки простої індикації стану системи. Наприклад, кольорове підсвічування може бути використане для навігації в будівлі, вказуючи користувачам шлях до певних зон або виходів у випадку надзвичайних ситуацій. Ця функція може бути особливо корисною в великих офісних комплексах, медичних установах або торгових центрах, де швидка та інтуїтивно зрозуміла навігація є критично важливою для забезпечення безпеки та комфорту відвідувачів.

Ще однією перевагою використання RGB світлодіодів є можливість їх інтеграції з іншими системами безпеки та автоматизації будівлі. Наприклад, індикація може бути синхронізована з системою пожежної сигналізації, змінюючи колір на червоний та активуючи режим блимання у випадку виникнення пожежної тривоги. Аналогічно, світлодіоди можуть взаємодіяти з системами клімат-контролю або освітлення, адаптуючи свою яскравість та колір відповідно до умов навколишнього середовища для забезпечення оптимальної видимості та комфорту користувачів.

З точки зору естетики та дизайну, RGB світлодіоди надають широкі можливості для створення візуально привабливих та сучасних систем контролю доступу. Можливість налаштування кольору та інтенсивності світіння дозволяє адаптувати зовнішній вигляд системи до загального дизайну інтер'єру або корпоративного стилю організації. Це особливо важливо для об'єктів, де естетичний аспект обладнання відіграє значну роль, наприклад, в готелях преміум-класу, мистецьких галереях або сучасних офісних просторах. Крім того, здатність RGB світлодіодів створювати динамічні світлові ефекти може бути використана для привернення уваги користувачів до важливих повідомлень або зміни в статусі системи.

Важливим аспектом при виборі RGB світлодіодів є їх екологічність та відповідність сучасним стандартам захисту навколишнього середовища. На відміну від деяких інших типів індикаторів, світлодіоди не містять шкідливих речовин, таких як ртуть, і споживають мінімальну кількість енергії, що сприяє зменшенню загального вуглецевого сліду системи. Це робить RGB світлодіоди ідеальним вибором для організацій, які прагнуть до впровадження екологічно чистих технологій та дотримання принципів сталого розвитку.

З фінансової точки зору, використання RGB світлодіодів в системах контролю доступу може призвести до значної економії коштів у довгостроковій перспективі. Хоча початкова вартість високоякісних RGB світлодіодів може бути дещо вищою порівняно з традиційними елементами індикації, їх тривалий термін служби, низьке енергоспоживання та мінімальні вимоги до обслуговування значно знижують загальну вартість володіння системою. Це робить RGB світлодіоди економічно ефективним вибором для організацій різного масштабу, від малих підприємств до великих корпорацій.

Окремо слід відзначити можливість програмного оновлення функціональності RGB світлодіодів без необхідності фізичної заміни компонентів. Це дозволяє адаптувати систему індикації до нових вимог безпеки або змін у політиці доступу організації шляхом простого оновлення

програмного забезпечення. Така гнучкість забезпечує довгострокову актуальність системи контролю доступу та захищає інвестиції організації в технологічну інфраструктуру.

У контексті психологічного впливу на користувачів, кольорова індикація, що забезпечується RGB світлодіодами, може відігравати важливу роль у створенні комфортного та безпечного середовища. Правильно підібрані кольори та режими індикації можуть сприяти зниженню стресу та напруги у користувачів системи контролю доступу, особливо в ситуаціях підвищеної безпеки. Наприклад, плавні переходи між кольорами можуть створювати більш заспокійливу атмосферу порівняно з різкими змінами або яскравим блиманням.

З точки зору інтеграції з сучасними технологіями, RGB світлодіоди [14] можуть бути ефективно використані в системах контролю доступу, що базуються на концепції Інтернету речей (IoT). Можливість дистанційного управління кольором та режимами роботи світлодіодів через мережу дозволяє реалізовувати складні сценарії взаємодії між різними системами безпеки та автоматизації будівлі. Наприклад, статус індикації може автоматично змінюватися залежно від даних, отриманих від інших IoT-пристроїв [15], таких як датчики руху, камери відеоспостереження або системи клімат-контролю.

Важливо також відзначити роль RGB світлодіодів у забезпеченні інклюзивності систем контролю доступу. Можливість налаштування яскравості та кольору індикації дозволяє адаптувати систему для користувачів з різними особливостями сприйняття, включаючи людей з порушеннями зору або колірного сприйняття. Це сприяє створенню більш доступного та комфортного середовища для всіх категорій користувачів, що є важливим аспектом сучасного підходу до проектування систем безпеки.

У контексті майбутнього розвитку технологій контролю доступу, використання RGB світлодіодів відкриває широкі можливості для впровадження інноваційних рішень. Наприклад, інтеграція з системами

штучного інтелекту та машинного навчання може дозволити створювати адаптивні схеми індикації, які автоматично налаштовуються під індивідуальні особливості та переваги кожного користувача. Це може включати персоналізовані кольорові схеми, оптимізовані для максимальної зрозумілості та комфорту сприйняття конкретним користувачем.

Таким чином, вибір RGB світлодіода в якості елемента індикації для комп'ютеризованої системи контролю доступу з хешуванням персональних даних є всебічно обґрунтованим рішенням, яке забезпечує оптимальне поєднання функціональності, енергоефективності, довговічності, безпеки та зручності використання. Цей вибір дозволяє створити інтуїтивно зрозумілу, гнучку та ефективну систему візуальної комунікації з користувачем, яка може бути легко адаптована до різноманітних вимог та умов експлуатації. Використання RGB світлодіодів не тільки підвищує загальну ефективність та надійність системи контролю доступу, але й створює основу для подальшого розвитку та вдосконалення технологій безпеки, відповідаючи сучасним тенденціям в галузі автоматизації, енергоефективності та захисту навколишнього середовища.

2.6 Обгрунтування вибору сервопривода MG996R як виконавчого механізму

При проектуванні комп'ютеризованої системи контролю доступу з хешуванням персональних даних вибір надійного та ефективного виконавчого механізму є критично важливим аспектом. У цьому контексті сервопривод MG996R був обраний як оптимальне рішення для реалізації фізичного контролю доступу [16], зокрема для управління дверними замками або турнікетами. Цей вибір базується на ряді ключових характеристик та переваг, які робить MG996R ідеальним кандидатом для даної системи.

Перш за все, MG996R відрізняється високим крутним моментом, який становить 9,4 кг*см при напрузі 4,8В та може досягати 11 кг*см при напрузі 6В. Така потужність забезпечує надійне та швидке відкриття і закриття дверей або управління турнікетом навіть в умовах значного навантаження, що є критично важливим для системи контролю доступу, яка повинна працювати безвідмовно в різних умовах експлуатації.

Крім того, MG996R характеризується високою точністю позиціонування, що дозволяє точно контролювати положення механізму замка або турнікету, забезпечуючи як надійне закриття, так і плавне відкриття без ривків або застрягань. Це особливо важливо для забезпечення комфорту користувачів та запобігання можливим пошкодженням механізму через некоректну роботу.

Важливою характеристикою MG996R є його швидкодія: сервопривод здатний повертатися на 60 градусів всього за 0,17 секунди, що забезпечує миттєву реакцію системи на команду відкриття або закриття. Така швидкість роботи є критичною для системи контролю доступу, де затримки можуть призвести до незручностей для користувачів або навіть до потенційних проблем безпеки.

MG996R також відрізняється високою надійністю та довговічністю завдяки використанню в його конструкції металевих шестерень. Це значно підвищує термін служби сервоприводу порівняно з моделями, що використовують пластикові шестерні, особливо в умовах інтенсивної експлуатації, характерної для систем контролю доступу. Металеві шестерні також забезпечують більш плавний рух та меншу схильність до зносу, що знижує необхідність в частому обслуговуванні або заміні.

З точки зору інтеграції в систему, MG996R є надзвичайно зручним завдяки стандартному інтерфейсу підключення, який сумісний з більшістю популярних мікроконтролерів, включаючи Arduino UNO, що використовується в даній системі. Сервопривод керується за допомогою PWM

(широтно-імпульсної модуляції) сигналу, що дозволяє легко контролювати його положення з високою точністю. Це спрощує процес програмування та налаштування системи, а також забезпечує гнучкість у реалізації різних алгоритмів управління доступом.

Важливо відзначити, що MG996R має компактні розміри (40,7 х 19,7 х 42,9 мм) та невелику вагу (55г), що дозволяє легко інтегрувати його в різні типи дверних конструкцій або турнікетів без необхідності значних модифікацій існуючої інфраструктури. Це особливо важливо при модернізації існуючих систем контролю доступу або при встановленні системи в приміщеннях з обмеженим простором.

Ще однією перевагою MG996R є його відносно низьке енергоспоживання, що важливо для забезпечення тривалої автономної роботи системи у випадку відключення основного живлення. При напрузі 4,8В сервопривод споживає близько 500мА під навантаженням, що дозволяє ефективно використовувати резервні джерела живлення, такі як акумуляторні батареї. Це підвищує надійність системи та забезпечує безперервність її роботи навіть в умовах нестабільного електропостачання.

З точки зору безпеки, важливо відзначити, що MG996R має вбудований захист від перевантаження, який запобігає пошкодженню сервоприводу у випадку застрягання механізму або спроби силового зламу. Це додатковий рівень захисту, який підвищує загальну надійність системи контролю доступу та зменшує ризик виходу з ладу критично важливих компонентів.

Крім того, MG996R має широкий діапазон робочих температур (від -30°C до +60°C), що дозволяє використовувати його як в приміщеннях, так і на вулиці, забезпечуючи стабільну роботу системи в різних кліматичних умовах. Це особливо важливо для об'єктів, де точки контролю доступу можуть бути розташовані як всередині будівлі, так і на зовнішніх входах.

При виборі MG996R (рис. 2.4) також враховувалася його доступність на ринку та наявність широкої підтримки з боку спільноти розробників. Це

забезпечує легкість отримання запасних частин, технічної документації та прикладів коду для інтеграції сервоприводу в різні системи. Така підтримка спрощує процес розробки, налагодження та обслуговування системи, що є важливим фактором для довгострокової експлуатації.

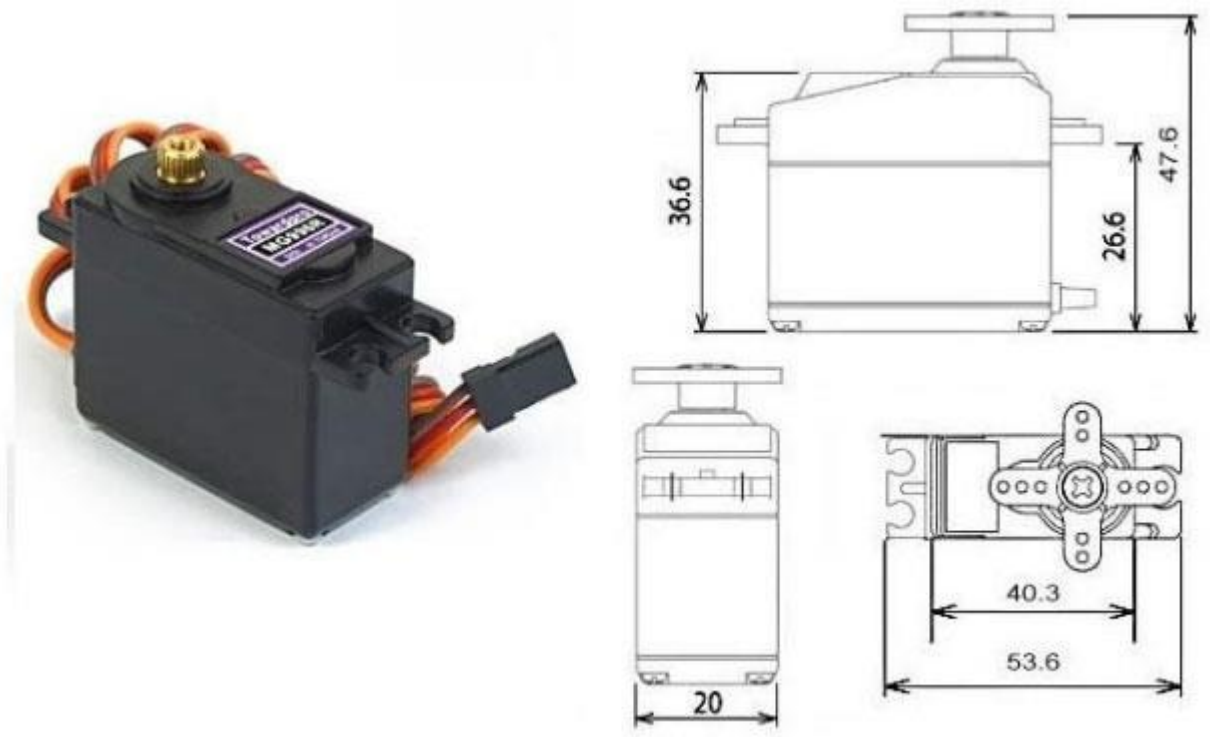

Рисунок 2.4 – Сервопривод MG996R

Варто також відзначити економічну ефективність MG996R. Незважаючи на свої високі технічні характеристики, цей сервопривод має відносно доступну ціну, що робить його оптимальним вибором для проектів різного масштабу - від невеликих офісів до великих корпоративних систем контролю доступу. Це дозволяє створювати економічно ефективні рішення без компромісів щодо якості та надійності. У контексті безпеки важливо відзначити, що MG996R забезпечує достатній рівень механічного опору для запобігання несанкціонованому доступу. Його потужний крутний момент та міцна конструкція створюють додатковий фізичний бар'єр, який складно подолати без спеціальних інструментів. Це доповнює електронні засоби

захисту системи, створюючи комплексний підхід до забезпечення безпеки об'єкта. З точки зору масштабованості системи, важливою перевагою MG996R є можливість легкого масштабування системи. При необхідності розширення системи контролю доступу на додаткові точки входу, можна легко додати нові сервоприводи, використовуючи стандартні інтерфейси підключення та наявні бібліотеки для управління.

Ці переваги та недоліки підкреслюють сильні сторони сервоприводу MG996R у термінах крутного моменту, точності, швидкості та надійності, а також відзначають його обмеження в термінах вартості, безперервної потужності та складності.

Це забезпечує гнучкість системи та можливість її адаптації до зростаючих потреб організації без необхідності повної заміни обладнання. Підсумовуючи, вибір сервоприводу MG996R як виконавчого механізму для комп'ютеризованої системи контролю доступу з хешуванням персональних даних є обґрунтованим та оптимальним рішенням. Його високий крутний момент, точність, надійність, швидкодія, компактність, енергоефективність та доступність роблять його ідеальним кандидатом для реалізації фізичного контролю доступу. MG996R забезпечує необхідний баланс між потужністю, надійністю та економічною ефективністю, що дозволяє створити високоякісну систему контролю доступу, здатну задовольнити вимоги різноманітних об'єктів та організацій.

2.7 Проектування схеми та embed-програмного забезпечення

Проектування схеми та embed-програмного забезпечення є ключовим етапом у розробці комп'ютеризованої системи контролю доступу на базі Arduino UNO. Цей процес включає в себе створення електричної схеми підключення всіх компонентів системи та розробку програмного коду для керування їх роботою. Електрична схема системи базується на

мікроконтролері Arduino UNO, до якого підключаються основні периферійні пристрої: RFID-зчитувач MFRC522, сервопривод MG996R та RGB світлодіод для індикації.

RFID-зчитувач MFRC522 підключається до Arduino UNO через SPI інтерфейс, використовуючи піни 10 (SDA), 11 (MOSI), 12 (MISO) та 13 (SCK). Додатково, пін RST модуля MFRC522 підключається до піну 9 Arduino для можливості програмного скидання зчитувача.

Сервопривод MG996R підключається до піну 3 Arduino UNO для керування його положенням. RGB світлодіод підключається до пінів 5 (червоний), 6 (зелений) та 7 (синій) Arduino UNO через резистори номіналом 220 Ом для обмеження струму.

Живлення всієї системи здійснюється від стабілізованого джерела з напругою 5В та струмом не менше 2А. Для захисту від перенапруг та електростатичних розрядів рекомендується встановити захисні діоди на вхідних лініях Arduino UNO.

Після завершення проектування електричної схеми, наступним кроком є розробка embed-програмного забезпечення для Arduino UNO. Програмний код реалізується в середовищі Arduino IDE на мові програмування C++. Основні функціональні блоки програми включають: ініціалізацію та налаштування всіх компонентів системи, зчитування даних з RFID-мітки, обробку отриманих даних та прийняття рішення про надання доступу, керування сервоприводом для відкриття/закриття дверей, керування RGB світлодіодом для індикації стану системи, взаємодію з комп'ютером через послідовний порт для передачі даних та отримання команд.

Систсема в головному циклі має постійно слухати команду від компютера через послідовний порт. Якщо команда "r" (тобто read) то система переходить в цикл постійної роботи з RFID модулем. Якщо користувач наблизив карту до зчитувача, то зчитати перші 64 символи із внутрішньої памяті RFID карти, передати на компютер і чекати відповіді до 5 секунд. Якщо

отримано відповідь d (тобто denied), то запалити червоний колір світлодіода і повернутися до циклу постійної роботи з RFID модулем. Якщо отримано відповідь g (тобто granted), то запалити зелений колір світлодіода, повернути сервопривід і повернутися до циклу постійної роботи з RFID модулем. Якщо отримано відповідь w (тобто w), то запалити червоний колір світлодіода і повернутися до циклу постійної роботи з RFID модулем. Якщо отримано відповідь w (тобто write), то прийняти 64 символи хеш коду з компютера і записати в память RFID карти, а тоді повернутися до головного циклу.

Лістинг коду для реалізації основних функцій системи контролю доступу на базі Arduino UNO наведено в додатках. Розглянемо детально кожен рядок наведеного програмного коду:

```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>
```

Ці рядки підключають необхідні бібліотеки: SPI для роботи з SPI-інтерфейсом, MFRC522 для роботи з RFID-модулем, та Servo для керування сервоприводом.

```
#define SS_PIN 10
#define RST_PIN 9
```

Визначаються константи для пінів SS (Slave Select) та RST (Reset) RFID-модуля.

```
MFRC522 mfrc522(SS_PIN, RST_PIN);
```

Створюється об'єкт mfrc522 класу MFRC522 для роботи з RFID-модулем.

```
Servo servo;
```

Створюється об'єкт servo класу Servo для керування сервоприводом.

```
  void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  servo.attach(3);
  pinMode(ledPin, OUTPUT);
}
```

Функція setup() ініціалізує послідовний порт, SPI-інтерфейс, RFID-модуль, прикріплює сервопривід до піну 3 та налаштовує пін світлодіода як вихід.

```
void loop() {
  char command;
  if (Serial.available()) {
    command = Serial.read();
    switch (command) {
      case 'r':
        readRFID();
        break;
      case 'w':
        writeRFID();
        break;
      default:
        break;
    }
  }
}
```

Головний цикл програми. Він перевіряє наявність команд від комп'ютера через послідовний порт і викликає відповідні функції для читання або запису RFID.

```
void readRFID() {
  if (mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial()) {
    String cardData = "";
    for (byte i = 0; i < 16; i++) {
      cardData += String(mfrc522.uid.uidByte[i], HEX);
    }
    Serial.println(cardData);
    delay(5000);
  } }
```

Функція readRFID() перевіряє наявність RFID-карти, зчитує її UID, перетворює його в рядок і відправляє на комп'ютер. Потім очікує 5 секунд на відповідь.

```
void writeRFID() {
  String hash = "";
  while (hash.length() < 64) {
    if (Serial.available()) {
      hash += Serial.read();
    }
  }
  for (byte i = 0; i < 16; i++) {
    mfrc522.uid.uidByte[i] = strtol(hash.substring(i * 4, (i
+ 1) * 4).c_str(), NULL, 16);
  }
  mfrc522.PICC_WriteCardSerial();
}
```

Функція writeRFID() отримує 64-символьний хеш-код з комп'ютера, розбиває його на 16 байтів і записує в пам'ять RFID-карти.

```
void accessGranted() {
  digitalWrite(ledPin, HIGH);
  servo.write(90);
  delay(5000);
  servo.write(0);
  digitalWrite(ledPin, LOW);
```

```
}
```

Функція writeRFID() отримує 64-символьний хеш-код з комп'ютера, розбиває його на 16 байтів і записує в пам'ять RFID-карти.

```
void accessGranted() {
  digitalWrite(ledPin, HIGH);
  servo.write(90);
  delay(5000);
  servo.write(0);
  digitalWrite(ledPin, LOW);
}
```

Функція accessGranted() вмикає світлодіод, відкриває двері (повертає сервопривід), чекає 5 секунд, закриває двері і вимикає світлодіод.

Апаратна частина системи в роботі зображена на рисунку 2.5.

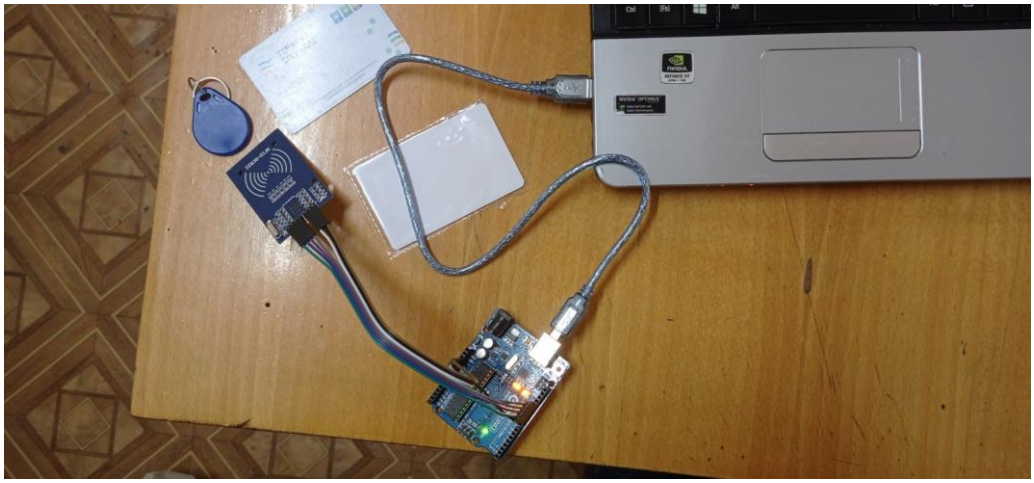

Рисунок 2.5 – Апаратна частина проектованої системи контролю доступу

Після завершення розробки та тестування embed-програмного забезпечення, необхідно створити детальну документацію, яка включатиме опис функціональності системи, інструкції з налаштування та експлуатації, а також рекомендації щодо технічного обслуговування та усунення можливих несправностей. Така документація значно полегшить процес впровадження та подальшого обслуговування системи.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ DESKTOP ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РОБОТИ З СИСТЕМОЮ КОНТРОЛЮ ДОСТУПУ

Розділ 3 присвячений реалізації desktop програмного забезпечення для роботи з системою контролю доступу. Це програмне забезпечення є комплексним рішенням, що поєднує потужний backend на мові Python та інтуїтивно зрозумілий frontend, реалізований з використанням сучасних веб-технологій. Ключовою особливістю системи є її тісна інтеграція з апаратною частиною через COM-порт, що забезпечує ефективний двосторонній обмін даними та дозволяє здійснювати централізоване керування всією системою контролю доступу.

Архітектура програмного забезпечення складається з кількох основних компонентів. Backend частина на Python включає REST API сервер, модуль роботи з базою даних SQLite, модуль комунікації з апаратною частиною через COM-порт та модуль обробки та хешування персональних даних. Frontend частина складається з інтерфейсу користувача на основі веб-технологій, модуля управління базою даних, модуля розрахунку та запису хеш-значень, а також модуля ручного управління доступом. Важливим компонентом системи є база даних SQLite. Така архітектура забезпечує високу гнучкість, масштабованість та ефективність системи, дозволяючи легко адаптувати її до різних умов експлуатації та розширювати функціональність без суттєвих змін в основній структурі.

Backend частина системи, реалізована на мові програмування Python, є ядром всього програмного комплексу. Вибір Python обумовлений його потужністю, гнучкістю та багатою екосистемою бібліотек, що дозволяє

ефективно вирішувати широкий спектр завдань в рамках системи контролю доступу. Центральним елементом backend частини є REST API сервер, розроблений з використанням фреймворку Flask. Цей сервер забезпечує взаємодію між різними компонентами системи, обробляє запити від клієнтської частини та керує всіма основними процесами, включаючи аутентифікацію користувачів, управління правами доступу та генерацію звітів.

Особлива увага в backend частині приділена забезпеченню безпеки персональних даних користувачів. Для цього реалізовано механізм хешування з використанням алгоритму SHA-256 з бібліотеки hashlib. Процес хешування відбувається наступним чином: особисті дані користувача об'єднуються в єдиний рядок, до отриманого рядка застосовується функція SHA-256, а результуюче хеш-значення зберігається в базі даних. Такий підхід забезпечує однонаправлене перетворення персональних даних, що унеможливлює їх відновлення з хеш-значення навіть у випадку несанкціонованого доступу до бази даних.

Для роботи з базою даних використовується SQLite - легковагове рішення, яке не потребує окремого сервера баз даних і зберігає всю інформацію у вигляді єдиного файлу. Це значно спрощує процеси резервного копіювання та відновлення системи, а також дозволяє легко переносити базу даних між різними середовищами. Структура бази даних включає дві основні таблиці: таблицю працівників та таблицю відвідуваності. Така структура дозволяє ефективно зберігати та управляти інформацією про користувачів та їх активність, забезпечуючи при цьому високий рівень захисту персональних даних.

Модуль комунікації з апаратною частиною через COM-порт реалізований з використанням бібліотеки pyserial. Цей модуль забезпечує надійний двосторонній обмін даними між програмним забезпеченням та

фізичними компонентами системи контролю доступу, такими як RFID-зчитувачі та електронні замки. Завдяки цьому модулю система може в реальному часі отримувати інформацію про спроби доступу, передавати команди на відкриття або закриття дверей, а також оновлювати налаштування апаратної частини.

Frontend частина системи реалізована з використанням сучасних веб-технологій: HTML5, CSS3 та Vanilla JavaScript. Такий підхід забезпечує кросплатформенність рішення, дозволяючи використовувати систему на різних пристроях без необхідності встановлення додаткового програмного забезпечення. Інтерфейс користувача розроблений з урахуванням принципів user-friendly дизайну та адаптивної верстки, що забезпечує зручність використання системи на пристроях з різними розмірами екранів.

Frontend частина має три основні режими роботи: режим управління базою даних, режим розрахунку та запису хеш-значень, а також режим ручного управління доступом. Режим управління базою даних надає адміністраторам повний контроль над інформацією в базі даних SQLite. Він дозволяє виконувати операції створення, читання, оновлення та видалення записів про користувачів. Крім того, в цьому режимі реалізовано функціонал для генерації статистики відвідуваності за певний період часу на основі другої таблиці SQLite бази даних.

Режим розрахунку та запису хеш-значень представляє собою спеціалізовану форму для роботи з персональними даними користувачів. Адміністратор може вводити персональні дані, система автоматично розраховує хеш-значення SHA-256 і зберігає його в базі даних. Також в цьому режимі реалізовано функціонал для управління PIN-кодами користувачів. Ця функція дозволяє створювати, змінювати та видаляти PIN-коди, які можуть використовуватися як додатковий фактор аутентифікації. Система

автоматично перевіряє унікальність PIN-кодів та їх відповідність встановленим політикам безпеки.

Режим ручного управління доступом забезпечує можливість авторизації користувачів у випадках, коли апаратна частина системи недоступна. Адміністратор може вручну ввести персональні дані та PIN-код користувача для перевірки прав доступу.

Кожен з цих режимів має інтуїтивно зрозумілий інтерфейс з відповідними формами введення даних, таблицями для відображення інформації та елементами управління. Інтерфейс розроблений з урахуванням принципів accessibility, що робить систему доступною для користувачів з різними потребами та можливостями. Крім того, реалізовано систему підказок та контекстної допомоги, яка полегшує освоєння системи новими користувачами та зменшує ймовірність помилок при роботі з системою.

Механізм резервного копіювання забезпечує регулярне створення резервних копій бази даних та конфігураційних файлів системи на окремий носій. Це дозволяє швидко відновити роботу системи у випадку технічних збоїв або інших непередбачених ситуацій. Крім того, реалізовано функціонал для шифрування резервних копій, що забезпечує додатковий рівень захисту даних.

Розроблена система має модульну архітектуру, що забезпечує її високу масштабованість та можливість легкого розширення функціональності. Потенційні напрямки розвитку системи включають інтеграцію додаткових методів аутентифікації, таких як біометричні дані або двофакторна аутентифікація, розширення функціональності звітності та аналітики, включаючи інтеграцію з системами бізнес-аналітики, розробку мобільного додатку для віддаленого управління системою, а також інтеграцію з іншими системами безпеки, такими як системи відеоспостереження або пожежної сигналізації.

Інтеграція біометричних методів аутентифікації може включати використання відбитків пальців, розпізнавання обличчя або сканування сітківки ока. Це дозволить ще більше підвищити рівень безпеки системи та зручність її використання. Розширення функціональності звітності та аналітики може включати розробку інтерактивних дашбордів для візуалізації ключових показників, впровадження алгоритмів машинного навчання для прогнозування паттернів відвідуваності та виявлення аномалій.

Розробка мобільного додатку дозволить адміністраторам та керівникам отримувати доступ до ключової інформації та управляти системою в режимі реального часу з будь-якого місця. Це особливо корисно для великих організацій з розподіленою структурою. Інтеграція з іншими системами безпеки дозволить створити єдиний центр управління безпекою об'єкта, що підвищить ефективність роботи служби безпеки та зменшить час реакції на потенційні загрози.

Таким чином, розроблене desktop програмне забезпечення для роботи з системою контролю доступу представляє собою комплексне, безпечне та гнучке рішення, яке може бути легко адаптоване до різних умов експлуатації та розширене відповідно до зростаючих потреб організації. Воно забезпечує ефективне управління доступом, захист персональних даних та надає широкі можливості для аналізу та оптимізації процесів безпеки.

3.1 Обгрунтування вибору механізму шифрування та контролю доступу

Вибір механізму шифрування та контролю доступу є фундаментальним аспектом розробки будь-якої системи безпеки, особливо коли йдеться про комп'ютеризовану систему контролю доступу з хешуванням персональних даних. Цей вибір не лише визначає рівень захисту системи від потенційних загроз, але й впливає на її продуктивність, масштабованість та зручність

використання. При розробці такої системи необхідно враховувати широкий спектр факторів, включаючи поточні та майбутні вимоги до безпеки, законодавчі норми, технологічні тенденції та специфічні потреби організації.

Вибір SHA-256 як алгоритму хешування є обґрунтованим рішенням, але варто детальніше розглянути його характеристики та альтернативи. SHA-256 належить до сімейства алгоритмів SHA-2, розроблених Національним інститутом стандартів і технологій США (NIST). Цей алгоритм широко використовується в різних сферах, від захисту паролів до створення цифрових підписів та блокчейн-технологій.

Основні переваги SHA-256:

1. Криптографічна стійкість: На даний момент не існує відомих практичних атак, які б дозволили знайти колізії або відновити вхідні дані з хеш-значення SHA-256.

2. Швидкість обчислень: SHA-256 оптимізований для 32-бітних систем, що забезпечує високу продуктивність на більшості сучасних комп'ютерних систем.

3. Фіксований розмір виходу: 256-бітний вихід забезпечує достатній рівень захисту від колізій, зберігаючи при цьому ефективність зберігання та обробки даних.

4. Широка підтримка: SHA-256 підтримується більшістю програмних бібліотек та апаратних засобів, що спрощує його інтеграцію в різні системи.

Однак, варто розглянути і потенційні альтернативи:

1. SHA-3: Новіше сімейство хеш-функцій, розроблене NIST як альтернатива SHA-2. SHA-3 має подібний рівень безпеки, але використовує інший алгоритм (Keccak), що може бути корисним для диверсифікації криптографічних примітивів у системі.

2. Blake2: Швидкий криптографічний хеш-алгоритм, який часто перевершує SHA-3 і SHA-256 за швидкодією, особливо на 64-бітних платформах.

3. Argon2: Спеціалізований алгоритм для хешування паролів, який розроблений для протидії атакам за допомогою спеціалізованого обладнання (наприклад, ASIC).

Вибір між цими алгоритмами залежить від конкретних вимог системи. Наприклад, якщо швидкодія є критично важливою, Blake2 може бути кращим вибором. Якщо система потребує додаткового захисту від атак з використанням спеціалізованого обладнання, Argon2 може бути більш підходящим для хешування паролів.

Реалізація механізму "солі" є важливим аспектом захисту від атак з використанням заздалегідь обчислених таблиць хешів. Проте, варто розглянути і додаткові методи посилення безпеки:

1. Перцювання (Peppering): Додавання секретного значення (перцю) до даних перед хешуванням. На відміну від солі, перець не зберігається разом з хешем, а є частиною конфігурації системи.

2. Ключове розтягування (Key stretching): Використання функцій, що вимагають значних обчислювальних ресурсів, таких як PBKDF2, bcrypt або scrypt, для створення хешів паролів. Це значно ускладнює проведення атак методом повного перебору.

3. Використання HMAC: Застосування хеш-функції з ключем (HMAC) може забезпечити додатковий рівень захисту, особливо при роботі з токенами аутентифікації.

Важливо також розглянути аспект оновлення системи безпеки. Криптографічні алгоритми можуть стати вразливими з часом через розвиток обчислювальних можливостей та появу нових методів криптоаналізу. Тому система повинна бути спроектована з можливістю легкого оновлення

алгоритмів хешування без необхідності повної реструктуризації бази даних або інфраструктури. В нашому випадку достатньо просто підмінити бібліотеку pyhash на більш нову і замінити врапери функцій.

Наприклад, можна реалізувати механізм версійності хешів, де кожен хеш зберігається разом з ідентифікатором використаного алгоритму. Це дозволить системі підтримувати кілька алгоритмів одночасно і поступово мігрувати на нові, більш безпечні алгоритми без переривання роботи системи.

У контексті захисту персональних даних важливо також розглянути концепцію мінімізації даних. Замість зберігання повних персональних даних у хешованому вигляді, можна розглянути можливість зберігання лише мінімально необхідної інформації. Наприклад, замість повного імені можна зберігати лише ініціали або унікальний ідентифікатор користувача. Це не тільки підвищує безпеку, але й відповідає принципам "privacy by design" та вимогам законодавства про захист персональних даних. В нашому випадку персональні дані не зберігаються зовсім, а унікальним ідентифікатором є 64-символьний хеш, що повністю виключає можливість витоку персональних даних.

Реалізація механізму хешування повинна також враховувати можливість майбутніх змін у законодавстві або галузевих стандартах. Наприклад, якщо в майбутньому будуть введені нові вимоги щодо довжини ключів шифрування або використання конкретних алгоритмів, система повинна бути достатньо гнучкою для адаптації до цих змін.

У наступній частині ми розглянемо більш детально аспекти вибору та реалізації моделі контролю доступу.

Вибір моделі контролю доступу є критично важливим аспектом розробки системи безпеки, оскільки вона визначає, як саме буде здійснюватися управління доступом до ресурсів та інформації [17]. Кожна модель має свої особливості, переваги та недоліки, які необхідно ретельно

проаналізувати в контексті конкретних вимог організації та специфіки системи.

Розглянемо детальніше кожну з основних моделей контролю доступу. Mandatory Access Control (MAC) є однією з найбільш строгих моделей контролю доступу. В цій моделі доступ контролюється централізовано на основі міток безпеки, присвоєних об'єктам та суб'єктам. MAC має високий рівень безпеки та контролю, ефективний захист від витоку інформації та відповідає вимогам високосекретних середовищ. Однак, ця модель має свої недоліки, такі як складність в управлінні та налаштуванні, низька гнучкість та може знизити продуктивність роботи користувачів. Прикладом застосування MAC є військові системи, де необхідно строго контролювати доступ до секретної інформації.

Discretionary Access Control (DAC) надає користувачам більше контролю над своїми ресурсами, дозволяючи їм самостійно встановлювати права доступу. DAC має переваги, такі як висока гнучкість, користувачі можуть ефективно керувати доступом до своїх ресурсів та проста в реалізації та розумінні. Однак, ця модель також має недоліки, такі як може призвести до непослідовності в політиках безпеки, ризик надмірного надання прав доступу та складність централізованого управління безпекою. Прикладом застосування DAC є файлові системи персональних комп'ютерів, де користувачі самі вирішують, кому надати доступ до своїх файлів.

Role Based Access Control (RBAC) базується на ролях, які призначаються користувачам. Права доступу пов'язані з ролями, а не з окремими користувачами. RBAC має переваги, такі як спрощене управління правами доступу, легко масштабується, відповідає організаційній структурі більшості підприємств та зменшує ризик людської помилки при призначенні прав. Однак, ця модель також має недоліки, такі як може бути складно реалізувати дуже специфічні права доступу та потребує ретельного планування ролей та їх

прав. Прикладом застосування RBAC є корпоративні системи управління, де права доступу визначаються посадою співробітника.

Rule Based Access Control (RB-RBAC) використовує набір правил для визначення доступу на основі атрибутів користувача, ресурсу та середовища. RB-RBAC має переваги, такі як висока гнучкість у визначенні умов доступу, можливість врахування контексту при прийнятті рішень про доступ та добре підходить для динамічних середовищ. Однак, ця модель також має недоліки, такі як складність в управлінні великою кількістю правил, потенційні конфлікти між правилами та може вимагати значних обчислювальних ресурсів для обробки складних правил. Прикладом застосування RB-RBAC є системи охорони здоров'я, де доступ до даних пацієнтів може залежати від багатьох факторів, включаючи роль лікаря, відділення, час доби тощо.

Вибір RBAC з трьома рівнями доступу для даної системи є обґрунтованим рішенням, яке забезпечує баланс між безпекою, гнучкістю та простотою управління.

У підсумку у цьому розділі детально розглядаються аспекти вибору механізмів шифрування та контролю доступу для комп'ютеризованої системи безпеки. У першій частині аналізується вибір алгоритму хешування, зокрема SHA-256, та його альтернативи. Обговорюються переваги SHA-256, такі як криптографічна стійкість, швидкість обчислень та широка підтримка. Також розглядаються альтернативні алгоритми, включаючи SHA-3, Blake2 та Argon2, кожен з яких має свої особливості та сфери застосування.

У цьому розділі підкреслюється важливість додаткових методів посилення безпеки, таких як використання солі, перцювання та ключове розтягування. Наголошується на необхідності проектування системи з можливістю оновлення алгоритмів шифрування без суттєвої реструктуризації. Також розглядається концепція мінімізації даних як спосіб підвищення безпеки та відповідності законодавству про захист персональних даних.

У другій частині аналізуються різні моделі контролю доступу, включаючи Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role Based Access Control (RBAC) та Rule Based Access Control (RB-RBAC). Кожна модель розглядається з точки зору її переваг, недоліків та типових сценаріїв застосування. Особлива увага приділяється RBAC як обраній моделі для системи, з пропозиціями щодо її вдосконалення.

У цьому розділі пропонуються різні способи покращення базової моделі RBAC, включаючи впровадження ієрархії ролей, застосування принципу найменших привілеїв, динамічне розділення обов'язків, використання тимчасових ролей та додавання елементів контекстно-залежного доступу. Також надаються рекомендації щодо практичної реалізації RBAC, такі як створення окремої бази даних прав доступу, кешування прав для підвищення продуктивності, детальне журналювання доступу та впровадження механізму делегування прав.

Загалом, у цьому розділі надається комплексний огляд ключових аспектів проектування системи безпеки, підкреслюючи важливість балансу між безпекою, гнучкістю та зручністю використання. Він також наголошує на необхідності враховувати майбутні зміни в технологіях та законодавстві при розробці системи безпеки.

3.2 Обгрунтування вибору Arduino IDE для апаратної частини системи

Arduino IDE було обрано для розробки програмного забезпечення апаратної частини системи контролю доступу з RFID, сервоприводом та RGB світлодіодом з ряду вагомих причин, які роблять це середовище розробки оптимальним вибором для даного проекту. Ширше розуміння переваг Arduino IDE та його функціональності дозволяє повністю оцінити доцільність його використання в контексті розробки систем контролю доступу.

Перш за все, Arduino IDE є безкоштовним та відкритим програмним забезпеченням, що робить його надзвичайно доступним для широкого кола розробників, від початківців до професіоналів. Ця особливість не тільки знижує вартість розробки проекту, але й сприяє формуванню активної спільноти користувачів, які постійно діляться досвідом, створюють нові бібліотеки та інструменти. В контексті системи контролю доступу це означає, що розробники можуть легко знайти рішення для специфічних проблем, пов'язаних з інтеграцією RFID-технологій, керуванням сервоприводами або налаштуванням складних схем індикації з використанням RGB світлодіодів.

Інтерфейс Arduino IDE розроблений з урахуванням потреб як новачків, так і досвідчених програмістів. Він надає простий та інтуїтивно зрозумілий спосіб написання, компіляції та завантаження коду на мікроконтролер Arduino. Ця особливість особливо важлива при розробці системи контролю доступу, де може бути необхідно швидко вносити зміни в код для адаптації до нових вимог безпеки або додавання нових функцій. Наприклад, якщо виникає потреба змінити логіку роботи сервоприводу для реалізації більш складного механізму блокування/розблокування, або модифікувати алгоритм зчитування RFID-міток для підтримки нового формату карт, Arduino IDE дозволяє зробити це швидко та ефективно.

Однією з ключових переваг Arduino IDE є наявність великої кількості вбудованих бібліотек та прикладів коду, які значно спрощують роботу з різноманітними компонентами та модулями. Для системи контролю доступу з використанням RFID, сервоприводу та RGB світлодіода особливо важливими є бібліотеки SPI, MFRC522 та Servo.

Бібліотека SPI (Serial Peripheral Interface) відіграє фундаментальну роль у забезпеченні комунікації між мікроконтролером Arduino та RFID-модулем. SPI - це синхронний послідовний протокол передачі даних, який забезпечує швидкий та надійний обмін інформацією між пристроями. У контексті

системи контролю доступу, ця бібліотека дозволяє налаштувати швидкі та стабільні комунікації з RFID-зчитувачем, що критично важливо для забезпечення миттєвої реакції системи на піднесення картки або мітки. SPI бібліотека надає можливість тонкого налаштування параметрів передачі даних, таких як швидкість, порядок бітів та режим роботи, що дозволяє оптимізувати роботу системи під конкретні вимоги проекту та характеристики використовуваного RFID-модуля.

Бібліотека MFRC522 спеціально розроблена для роботи з RFID-модулем RC522, який є одним з найпопулярніших рішень для систем контролю доступу малого та середнього масштабу. Ця бібліотека надає розробникам потужний набір інструментів для взаємодії з RFID-мітками. Вона включає функції для ініціалізації модуля, виявлення присутності RFID-карток, зчитування їх унікальних ідентифікаторів, а також читання та запису даних у різні сектори пам'яті RFID-міток. У контексті системи контролю доступу це дозволяє реалізувати такі важливі функції, як: перевірка авторизації користувача шляхом порівняння зчитаного ідентифікатора з базою даних дозволених карток; зберігання додаткової інформації про користувача безпосередньо на картці (наприклад, рівень доступу або термін дії перепустки); реалізація складних схем шифрування для підвищення безпеки системи. Крім того, бібліотека MFRC522 надає можливість роботи з різними типами RFID-міток, що робить систему більш гнучкою та адаптивною до різних сценаріїв використання.

Бібліотека Servo є незамінним інструментом для керування сервоприводом, який в системі контролю доступу зазвичай використовується як механізм блокування та розблокування дверей або турнікетів. Ця бібліотека значно спрощує процес управління сервоприводом, надаючи прості у використанні функції для встановлення кута повороту, контролю швидкості руху та обмеження діапазону руху сервоприводу. У контексті системи

контролю доступу це дозволяє реалізувати плавне та точне керування механізмом замка, що важливо не тільки для забезпечення надійності системи, але й для підвищення комфорту користувачів. Наприклад, можна запрограмувати поступове відкриття замка для зменшення шуму та зносу механізму, або реалізувати складні послідовності рухів для багатоступеневих систем блокування. Крім того, бібліотека Servo дозволяє легко налаштувати систему на роботу з різними моделями сервоприводів, що може бути корисно при модернізації або масштабуванні системи контролю доступу.

Хоча для керування RGB світлодіодом не вказана окрема бібліотека, Arduino IDE надає вбудовані функції для роботи з цифровими та аналоговими виходами, які можуть бути ефективно використані для управління кольором та яскравістю світлодіода. Це дозволяє реалізувати різноманітні схеми індикації статусу системи, що є важливим елементом інтерфейсу користувача в системі контролю доступу. Наприклад, зелений колір може сигналізувати про успішну аутентифікацію та надання доступу, червоний - про відмову у доступі або спробу несанкціонованого входу, а синій - про режим очікування або процес зчитування картки. Крім того, використовуючи функції ШІМ (широтно-імпульсної модуляції), доступні в Arduino IDE, можна створювати складні світлові ефекти, такі як плавна зміна кольорів або миготіння, що може бути використано для додаткової візуальної сигналізації різних станів системи або привертання уваги до важливих подій.

Arduino IDE також підтримує серійний монітор, який є надзвичайно корисним інструментом для налагодження та моніторингу роботи системи контролю доступу. Цей інструмент дозволяє в режимі реального часу відображати різноманітну інформацію, таку як статус зчитування RFID-карти, результати аутентифікації, дані про спроби доступу, стан сервоприводу та інші важливі параметри системи. У процесі розробки та тестування системи контролю доступу серійний монітор стає незамінним засобом для виявлення

та усунення помилок, оптимізації роботи системи та збору статистичних даних про її функціонування. Наприклад, через серійний монітор можна легко відстежити послідовність дій системи при спробі доступу, перевірити правильність зчитування даних з RFID-карток, або моніторити час відгуку системи на різні події.

Важливою перевагою Arduino IDE є його кросплатформеність. Це середовище розробки доступне для основних операційних систем, включаючи Windows, macOS та Linux, що забезпечує розробникам гнучкість у виборі робочого середовища. Ця особливість особливо цінна в контексті розробки систем контролю доступу, де може бути необхідно працювати з різними платформами в залежності від вимог проекту або переваг команди розробників. Крім того, кросплатформеність Arduino IDE спрощує процес розгортання та обслуговування систем контролю доступу в різних організаціях, де можуть використовуватися різні операційні системи.

Arduino IDE має вбудований компілятор, який автоматично обробляє багато низькорівневих деталей програмування мікроконтролера. Це дозволяє розробникам зосередитися на логіці роботи системи контролю доступу, не витрачаючи час на тонкощі взаємодії з апаратною частиною. Наприклад, компілятор автоматично оптимізує код для ефективного використання обмежених ресурсів мікроконтролера, що особливо важливо для систем контролю доступу, де швидкість реакції та надійність роботи є критичними факторами. Крім того, компілятор Arduino IDE здатний виявляти та повідомляти про потенційні проблеми в коді, такі як переповнення пам'яті або некоректне використання типів даних, що допомагає запобігти багатьом помилкам ще на етапі розробки.

Функція автоматичного форматування коду в Arduino IDE є ще однією корисною особливістю, яка допомагає підтримувати код чистим та легким для читання. Це особливо важливо при розробці складних систем контролю

доступу, де код може бути досить об'ємним і включати багато різних функціональних модулів. Автоматичне форматування забезпечує уніфікований стиль коду, що полегшує його розуміння та модифікацію, особливо при роботі в команді або при необхідності повернутися до проекту після тривалого періоду. Наприклад, це може бути критично важливим при додаванні нових функцій до існуючої системи контролю доступу або при передачі проекту іншому розробнику для подальшої підтримки та вдосконалення.

Arduino IDE постійно оновлюється та вдосконалюється завдяки активній спільноті розробників. Це забезпечує регулярне додавання нових функцій, виправлення помилок та покращення продуктивності середовища розробки. Для систем контролю доступу це означає можливість постійного вдосконалення та адаптації до нових технологій та вимог безпеки. Наприклад, з кожним оновленням можуть з'являтися нові бібліотеки або інструменти, які дозволяють реалізувати більш складні алгоритми шифрування, підтримку нових типів RFID-міток або інтеграцію з сучасними системами моніторингу та управління.

Велика спільнота користувачів Arduino означає, що існує багато форумів, туторіалів та ресурсів для вирішення потенційних проблем та обміну досвідом. Це особливо цінно при розробці систем контролю доступу, де можуть виникати специфічні технічні проблеми або питання щодо оптимізації роботи системи. Розробники можуть легко знайти відповіді на свої питання, поділитися власним досвідом або навіть знайти готові рішення для типових задач, що значно прискорює процес розробки та покращує якість кінцевого продукту.

Arduino IDE також підтримує розширення функціональності через систему плагінів. Це дозволяє додавати нові інструменти та можливості, специфічні для розробки систем контролю доступу. Наприклад, можна

встановити плагіни для більш зручної роботи з RFID-модулями, інструменти для аналізу продуктивності системи або засоби для автоматизованого тестування різних сценаріїв доступу.

Важливо відзначити, що Arduino IDE підтримує роботу з різними моделями плат Arduino та сумісними мікроконтролерами. Це дає можливість легко масштабувати систему контролю доступу, переходячи на більш потужні платформи при необхідності обробки більшої кількості точок доступу або реалізації більш складних алгоритмів безпеки. Наприклад, можна почати розробку на базовій платі Arduino UNO для прототипування, а потім легко перенести проект на більш потужну плату Arduino Mega або навіть на промисловий контролер, сумісний з Arduino.

Arduino IDE також надає можливість роботи з зовнішніми програматорами, що може бути корисно для оптимізації використання пам'яті мікроконтролера в системах контролю доступу. Це дозволяє завантажувати програму безпосередньо в пам'ять мікроконтролера, минаючи завантажувач Arduino, що звільняє додатковий простір для коду та даних. У контексті систем контролю доступу це може бути критично важливим для реалізації більш складних алгоритмів шифрування або зберігання більшої кількості записів про користувачів та події доступу.

Ще однією важливою перевагою Arduino IDE є можливість інтеграції з іншими інструментами розробки та системами контролю версій. Це особливо цінно при розробці складних систем контролю доступу, де може бути задіяно кілька розробників або потрібно вести ретельний облік змін у проекті. Наприклад, Arduino IDE можна легко інтегрувати з такими системами контролю версій як Git, що дозволяє ефективно відстежувати зміни в коді, створювати різні гілки розробки для експериментальних функцій та легко повертатися до попередніх версій системи у разі виникнення проблем.

Підтримка зовнішніх бібліотек в Arduino IDE також є значною перевагою при розробці систем контролю доступу. Окрім вбудованих бібліотек, розробники мають доступ до величезної кількості сторонніх бібліотек, створених спільнотою. Це дозволяє легко додавати нові функції до системи без необхідності розробляти їх з нуля. Наприклад, існують спеціалізовані бібліотеки для роботи з різними типами дисплеїв, які можуть бути використані для створення інформативного інтерфейсу користувача в системі контролю доступу. Також доступні бібліотеки для роботи з різними протоколами зв'язку, що може бути корисно при інтеграції системи контролю доступу з іншими системами безпеки або управління будівлею.

Arduino IDE також надає зручні інструменти для налагодження коду, такі як точки зупинки та покрокове виконання програми. Хоча ці функції обмежені порівняно з більш просунутими середовищами розробки, вони все ж надають цінні можливості для виявлення та виправлення помилок у логіці роботи системи контролю доступу. Наприклад, розробник може встановити точку зупинки на момент зчитування RFID-картки і покроково пройти процес аутентифікації, щоб переконатися у правильності роботи алгоритму.

Важливою особливістю Arduino IDE є підтримка бібліотек для роботи з різними протоколами зв'язку, такими як I2C, SPI, UART. Це дозволяє легко інтегрувати в систему контролю доступу додаткові компоненти, такі як зовнішні модулі пам'яті для зберігання великої кількості записів про користувачів, дисплеї для відображення інформації, або модулі бездротового зв'язку для передачі даних на центральний сервер. Наприклад, використовуючи протокол I2C, можна підключити EEPROM великої ємності для зберігання розширеної бази даних користувачів, що особливо корисно для систем з великою кількістю точок доступу.

Arduino IDE також підтримує роботу з аналоговими сенсорами, що може бути використано для додаткових функцій безпеки в системі контролю

доступу. Наприклад, можна інтегрувати датчики температури для моніторингу умов середовища в критичних зонах, або датчики тиску для виявлення спроб силового проникнення. IDE надає зручні функції для зчитування та обробки аналогових сигналів, що спрощує інтеграцію таких додаткових компонентів у систему.

Ще одним важливим аспектом є можливість використання Arduino IDE для програмування мікроконтролерів інших виробників, які сумісні з Arduino. Це дає розробникам систем контролю доступу більшу гнучкість у виборі апаратної платформи, дозволяючи використовувати спеціалізовані мікроконтролери з підвищеною безпекою або продуктивністю, зберігаючи при цьому звичне середовище розробки та існуючий код.

Arduino IDE також підтримує можливість створення та використання власних бібліотек. Це особливо корисно при розробці складних систем контролю доступу, де може бути доцільно виділити певні функції (наприклад, алгоритми шифрування або протоколи комунікації) в окремі бібліотеки для полегшення їх повторного використання та підтримки. Створення власних бібліотек також сприяє модульності коду, що покращує його читабельність та спрощує подальшу модифікацію системи.

Варто відзначити, що Arduino IDE підтримує функцію автоматичного завантаження драйверів для більшості популярних плат Arduino та сумісних пристроїв. Це значно спрощує процес налаштування середовища розробки, особливо для початківців або при роботі з новими типами плат. У контексті розробки систем контролю доступу це означає, що розробники можуть легко експериментувати з різними апаратними платформами, не витрачаючи час на складні процедури налаштування драйверів.

Arduino IDE також включає в себе менеджер плат, який дозволяє легко додавати підтримку нових типів плат та мікроконтролерів. Це особливо корисно при розробці систем контролю доступу, які можуть вимагати

специфічних апаратних рішень. Наприклад, якщо проект вимагає використання мікроконтролера з підвищеною безпекою або специфічними периферійними пристроями, розробник може легко додати підтримку такої плати через менеджер плат і продовжити роботу в звичному середовищі Arduino IDE.

Важливою особливістю Arduino IDE є підтримка багатьох мов програмування, включаючи C++ та асемблер. Хоча більшість проектів на Arduino розробляється з використанням спрощеного варіанту C++, можливість використання повного C++ або навіть вставок асемблерного коду дозволяє оптимізувати критичні ділянки програми для досягнення максимальної продуктивності. У контексті систем контролю доступу це може бути корисно, наприклад, для оптимізації алгоритмів шифрування або для забезпечення швидкої реакції системи на події.

Arduino IDE також надає можливість використання прошивок зі сторонніх джерел. Це дозволяє розробникам систем контролю доступу використовувати спеціалізовані прошивки, оптимізовані для конкретних завдань безпеки або продуктивності. Наприклад, існують прошивки, які забезпечують підвищений рівень захисту від несанкціонованого доступу до мікроконтролера, що може бути критично важливим для систем контролю доступу високого рівня безпеки.

Ще однією корисною функцією Arduino IDE є можливість налаштування параметрів компіляції. Це дозволяє розробникам оптимізувати код для конкретної апаратної платформи або специфічних вимог проекту. Наприклад, можна налаштувати рівень оптимізації компілятора для досягнення балансу між розміром коду та швидкістю його виконання, що може бути важливим для систем контролю доступу, які працюють на мікроконтролерах з обмеженими ресурсами.

Arduino IDE також підтримує функцію експорту скомпільованого бінарного файлу. Це дозволяє розробникам систем контролю доступу створювати прошивки, які можуть бути легко розповсюджені та встановлені на інші пристрої без необхідності повторної компіляції. Ця функція особливо корисна при масовому виробництві систем контролю доступу або при оновленні вже встановлених систем.

Нарешті, Arduino IDE має вбудовану систему документації та довідки, яка надає швидкий доступ до інформації про функції, константи та синтаксис мови програмування Arduino. Це особливо корисно при роботі зі специфічними функціями або бібліотеками, які використовуються в системах контролю доступу. Розробники можуть швидко отримати необхідну інформацію без необхідності шукати її в зовнішніх джерелах, що прискорює процес розробки та зменшує кількість помилок.

Таким чином, вибір Arduino IDE для розробки апаратної частини системи контролю доступу з RFID, сервоприводом та RGB світлодіодом є обґрунтованим та ефективним рішенням. Це середовище розробки надає всі необхідні інструменти та функції для створення надійної, безпечної та функціональної системи, забезпечуючи при цьому простоту використання, гнучкість та широкі можливості для оптимізації та вдосконалення проекту.

3.3 Обгрунтування вибору Notepad++ для технологій python та WEB

У процесі розробки комп'ютеризованої системи контролю доступу з хешуванням персональних даних, вибір оптимального середовища розробки є критично важливим фактором, який безпосередньо впливає на ефективність процесу створення програмного забезпечення, якість кінцевого продукту та можливості подальшої підтримки і розвитку проекту. Після ретельного аналізу доступних інструментів та врахування специфіки проекту, який включає в

себе як серверну частину на основі Python/Flask, так і клієнтську частину з використанням веб-технологій (HTML, CSS та Vanilla JavaScript), було прийнято рішення обрати Notepad++ як основний інструмент розробки. Це рішення базується на ряді суттєвих переваг, які надає Notepad++ для роботи з різними мовами програмування та технологіями як середовище програмування у широкому спектрі галузей.

Notepad++ є потужним, безкоштовним редактором вихідного коду з відкритим вихідним кодом, який має довгу історію розвитку та постійного вдосконалення. Його створення датується 2003 роком, і за цей час він став одним з найпопулярніших текстових редакторів серед розробників по всьому світу. Ключовою особливістю Notepad++ є його здатність підтримувати велику кількість мов програмування, включаючи Python, HTML, CSS та JavaScript, що робить його універсальним інструментом для різнопланових проектів. Ця універсальність є особливо цінною для нашого проекту, який охоплює як серверну, так і клієнтську розробку, оскільки дозволяє використовувати єдине середовище для роботи з усіма компонентами системи.

Одна з найбільш вагомих переваг Notepad++ - це його виняткова легкість та висока швидкодія. На відміну від повномасштабних інтегрованих середовищ розробки (IDE), які часто вимагають значних системних ресурсів та мають тривалий час завантаження, Notepad++ стартує практично миттєво та споживає мінімальну кількість оперативної пам'яті та процесорного часу. Ця особливість набуває особливого значення при роботі з проектом, який включає в себе різні технології та може вимагати одночасного запуску декількох інструментів розробки, серверів та баз даних. Використання Notepad++ дозволяє ефективно розподіляти ресурси комп'ютера, забезпечуючи швидку та плавну роботу всіх компонентів розробки без необхідності в потужному апаратному забезпеченні.

Notepad++ надає розширені можливості редагування тексту, які значно підвищують продуктивність розробника. Однією з ключових функцій є інтелектуальне підсвічування синтаксису, яке адаптується до різних мов програмування, включаючи Python, HTML, CSS та JavaScript. Ця функція не лише полегшує читання та розуміння коду, але й допомагає візуально виявляти синтаксичні помилки на ранніх етапах розробки, що суттєво зменшує час на відладку та покращує загальну якість кодової бази. Крім того, Notepad++ пропонує функцію автодоповнення коду, яка доступна для всіх підтримуваних мов. Ця функція значно прискорює процес написання коду, пропонуючи контекстно-залежні підказки та автоматично завершуючи часто використовувані конструкції. Це особливо корисно при роботі з Python API та HTML/CSS/JavaScript структурами, де правильне написання імен функцій, методів та атрибутів є критичним для коректної роботи програми.

Система плагінів є однією з найсильніших сторін Notepad++, яка дозволяє розширювати функціональність редактора відповідно до специфічних потреб проекту. Ця гнучкість особливо важлива для нашої системи контролю доступу, де вимоги можуть змінюватися в процесі розробки. Для роботи з серверною частиною на Python/Flask можна встановити ряд корисних плагінів, таких як PyNPP для розширеної підтримки Python, включаючи перевірку синтаксису в реальному часі, автоматичне форматування коду згідно з PEP 8 (стандарт стилю коду Python), та інтеграцію з віртуальними середовищами Python. Плагін NppExec дозволяє виконувати Python-скрипти безпосередньо з інтерфейсу Notepad++, що прискорює процес тестування окремих компонентів API. Для веб-розробки доступні плагіни, які значно покращують роботу з HTML, CSS та JavaScript. Наприклад, плагін "HTML Tag" автоматично закриває HTML-теги, "JSLint" виконує статичний аналіз JavaScript-коду для виявлення потенційних проблем, а "CSS Formatter" допомагає підтримувати чистоту та читабельність CSS-файлів. Крім того,

плагіни для інтеграції з системами контролю версій, такі як "NppGit" для Git, дозволяють ефективно керувати версіями коду безпосередньо з інтерфейсу редактора.

Вбудована підтримка регулярних виразів у Notepad++ є потужним інструментом, який значно розширює можливості обробки текстових даних. Ця функція особливо корисна при роботі з великими обсягами коду, конфігураційними файлами або під час масштабного рефакторингу. У контексті нашої системи контролю доступу, регулярні вирази можуть бути використані для швидкого пошуку та модифікації шаблонів у коді, наприклад, для оновлення форматів логування, зміни структури API-запитів або оптимізації HTML-шаблонів. Можливість виконувати складні операції пошуку та заміни з використанням регулярних виразів дозволяє автоматизувати багато рутинних завдань, значно прискорюючи процес розробки та підтримки проекту.

Здатність Notepad++ працювати з декількома документами одночасно в режимі вкладок або розділеного екрану є неоціненною перевагою для проекту, який охоплює різні технології. Розробник може легко переключатися між файлами Python для серверної логіки, HTML шаблонами, CSS стилями та JavaScript скриптами, зберігаючи контекст роботи та забезпечуючи цілісний підхід до розробки всієї системи. Ця функція особливо корисна при реалізації функціональності, яка вимагає одночасних змін у різних компонентах системи, наприклад, при додаванні нового API-ендпоінту, який потребує оновлення серверного коду, клієнтського JavaScript та HTML-шаблону.

Важливою перевагою Notepad++, яка часто недооцінюється, є його портативність. Програму можна встановити на USB-накопичувач та використовувати на різних комп'ютерах без необхідності повторної інсталяції та налаштування. Ця особливість надає значну гнучкість розробникам, дозволяючи їм працювати над проектом з різних робочих станцій, зберігаючи

при цьому всі свої налаштування, плагіни та макроси. У контексті розробки системи контролю доступу, де можуть бути задіяні різні спеціалісти (back-end розробники, front-end розробники, спеціалісти з безпеки), портативність Notepad++ забезпечує уніфіковане середовище розробки, яке легко переноситься між різними робочими місцями без втрати продуктивності.

Notepad++ має велику та активну спільноту користувачів та розробників, що є гарантією постійної підтримки та вдосконалення програми. Це критично важливо для довгострокових проектів, таких як наша система контролю доступу, оскільки забезпечує актуальність та сумісність редактора з новими версіями мов програмування та технологій, які використовуються в проекті. Активна спільнота також означає наявність великої бази знань, форумів та ресурсів, де можна знайти рішення практично для будь-якої проблеми або отримати пораду щодо оптимального використання Notepad++ в контексті розробки. Це суттєво скорочує час на вирішення технічних питань та сприяє постійному підвищенню кваліфікації розробників.

У контексті розробки REST API сервера на основі Python/Flask, Notepad++ надає всі необхідні інструменти для ефективного написання та редагування Python-коду. Підтримка синтаксису Python включає не лише базове підсвічування, але й розпізнавання специфічних для Flask конструкцій, що полегшує роботу з фреймворком. Автодоповнення коду в Notepad++ розуміє контекст Python-проектів, пропонуючи релевантні методи та атрибути при роботі з об'єктами Flask, що значно прискорює процес розробки API. Функція "Function List" дозволяє швидко навігувати між різними функціями та класами в Python-файлах, що особливо корисно при роботі з великими модулями.

Для роботи з базами даних, які є невід'ємною частиною систем контролю доступу, Notepad++ пропонує зручні інструменти для написання та редагування SQL-запитів. Хоча редактор не має вбудованого функціоналу для

прямого підключення до баз даних, а у випадку з SQLite, що не є серверною базою даних, він забезпечує відмінну підтримку синтаксису SQL, що полегшує розробку та налагодження запитів. У поєднанні з плагіном "SQLinForm", який форматує SQL-код для кращої читабельності, Notepad++ стає потужним інструментом для роботи з базами даних в контексті розробки API.

Для реалізації механізмів безпеки, зокрема хешування персональних даних, Notepad++ надає зручне середовище для роботи з криптографічними бібліотеками Python. Підсвічування синтаксису допомагає візуально відрізняти криптографічні функції та їх параметри, що зменшує ймовірність помилок при реалізації чутливих до безпеки компонентів системи. Крім того, можливість швидкого порівняння різних версій файлів (через вбудовану функцію порівняння або плагін "Compare") є неоціненною при аудиті змін у критичних з точки зору безпеки частинах коду.

Для розробки клієнтської частини (FrontEnd) з використанням HTML, CSS та Vanilla JavaScript, Notepad++ пропонує ряд функцій, які значно підвищують продуктивність веб-розробки. Підсвічування синтаксису для HTML автоматично виділяє теги, атрибути та їх значення різними кольорами, що покращує читабельність коду та допомагає швидко виявляти помилки в структурі документа. Для CSS Notepad++ надає не тільки підсвічування синтаксису, але й автодоповнення властивостей та їх значень, що прискорює процес стилізації елементів інтерфейсу. При роботі з JavaScript, редактор забезпечує інтелектуальне автодоповнення, включаючи методи об'єктів DOM та вбудовані функції, що особливо корисно при розробці інтерактивних елементів інтерфейсу користувача.

Функція "Folder as Workspace" в Notepad++ дозволяє організувати всі файли проекту в зручну деревоподібну структуру, що особливо важливо для великих веб-проектів з багатьма HTML, CSS та JavaScript файлами. Це забезпечує швидку навігацію між різними компонентами фронтенду,

дозволяючи розробнику легко переключатися між роботою над структурою, стилями та функціональністю веб-інтерфейсу системи контролю доступу.

Для тестування та налагодження клієнтської частини, Notepad++ може бути ефективно використаний у поєднанні з веб-браузерами та їх інструментами розробника. Можливість швидкого збереження змін в файлах та миттєвого оновлення сторінки в браузері створює ефективний цикл розробки. Крім того, плагін "Preview HTML" дозволяє попередньо переглядати HTML-файли безпосередньо в Notepad++, що корисно для швидкої перевірки базової структури та стилів без необхідності перемикання на браузер.

У контексті оптимізації продуктивності веб-інтерфейсу, Notepad++ надає інструменти для роботи з мініфікацією коду. Хоча редактор не має вбудованих функцій мініфікації, існують плагіни, які дозволяють автоматично стискати CSS та JavaScript файли, що важливо для покращення швидкості завантаження веб-сторінок системи контролю доступу.

Крос-платформна сумісність Notepad++ є ще однією важливою перевагою для нашого проекту. Хоча програма спочатку була розроблена для Windows, існують способи запустити її на Linux та macOS за допомогою Wine або альтернативних рішень. Це забезпечує гнучкість для розробників, які можуть працювати на різних операційних системах, зберігаючи при цьому єдине середовище розробки для всієї команди.

Вбудована підтримка кодування UTF-8 у Notepad++ особливо важлива при роботі з міжнародними проектами або системами, які повинні підтримувати багатомовність. Це гарантує, що всі символи, включаючи кирилицю та інші нелатинські алфавіти, будуть коректно відображатися та оброблятися без проблем з кодуванням.

Функція макросів у Notepad++ дозволяє автоматизувати повторювані завдання, що може значно підвищити продуктивність при роботі над великими

проектами. Наприклад, можна створити макрос для автоматичного форматування коду відповідно до стандартів проекту або для вставки часто використовуваних шаблонів коду.

Notepad++ також пропонує функцію "Пошук у файлах", яка дозволяє швидко знаходити потрібну інформацію у всьому проекті. Це особливо корисно при роботі з великою кодовою базою, коли потрібно швидко знайти використання певної функції або змінної.

Для забезпечення безпеки розробки, особливо важливої для системи контролю доступу, Notepad++ підтримує роботу з SSH через плагіни. Це дозволяє безпечно редагувати файли на віддалених серверах, що може бути критично важливим при налаштуванні та обслуговуванні системи в продакшн-середовищі.

Функція порівняння файлів у Notepad++ допомагає ефективно відстежувати зміни в коді та конфігураційних файлах. Це особливо корисно при роботі в команді або при налагодженні системи, коли потрібно швидко визначити, які саме зміни були внесені та як вони впливають на роботу системи.

Підтримка FTP/SFTP через плагіни дозволяє розробникам напряму редагувати файли на віддалених серверах, що спрощує процес розгортання та оновлення системи. Це особливо зручно при роботі з веб-серверами або при необхідності швидко внести зміни в продакшн-середовище.

Для розробки REST API на Python/Flask, Notepad++ надає зручні інструменти для роботи з JSON та XML, які використовуються в API-запитах та відповідях. Підсвічування синтаксису та валідація структури цих форматів даних допомагає уникнути помилок при розробці інтерфейсів API.

При роботі з фронтенд-частиною проекту, Notepad++ пропонує зручні інструменти для роботи з CSS, включаючи підсвічування кольорів та

автодоповнення властивостей. Це значно прискорює процес стилізації веб-інтерфейсу системи контролю доступу.

Варто зазначити, що Notepad++ регулярно оновлюється, забезпечуючи підтримку нових версій мов програмування та технологій. Це гарантує, що середовище розробки залишатиметься актуальним протягом усього життєвого циклу проекту.

У контексті безпеки, яка є критично важливою для системи контролю доступу з хешуванням персональних даних, Notepad++ не зберігає чутливу інформацію та не відправляє дані на зовнішні сервери. Це мінімізує ризики витоку конфіденційної інформації під час розробки.

Загалом, вибір Notepad++ як основного інструменту розробки для нашої комп'ютеризованої системи контролю доступу є обґрунтованим рішенням, яке забезпечує оптимальний баланс між функціональністю, продуктивністю та зручністю використання для всіх аспектів проекту, від серверної розробки на Python до створення клієнтського інтерфейсу з використанням веб-технологій.

3.4 Обгрунтування вибору структури бази даних

Обґрунтування вибору структури бази даних для системи контролю доступу з REST API сервером на основі Python/Flask є фундаментальним етапом у процесі проектування та розробки. Цей вибір має критичне значення, оскільки від нього залежить не лише ефективність та продуктивність системи, але й її безпека, масштабованість, та здатність адаптуватися до майбутніх вимог та технологічних змін. У контексті даної системи, після ретельного аналізу різноманітних варіантів та врахування специфічних вимог проекту, було прийнято рішення використовувати SQLite як основну систему управління базами даних (СУБД). Цей вибір обумовлений цілим рядом

ключових факторів, які в сукупності роблять SQLite оптимальним рішенням для реалізації даного проекту.

Перш за все, SQLite є вбудованою базою даних, яка не вимагає окремого серверного процесу або системи. Це надзвичайно важлива характеристика, яка має ряд суттєвих переваг у контексті системи контролю доступу. По-перше, це значно спрощує процес розгортання та обслуговування системи, особливо в умовах обмежених ресурсів або при необхідності швидкого розгортання на нових локаціях. Вся база даних зберігається в єдиному файлі, що робить її надзвичайно портативною і дозволяє легко створювати резервні копії або переносити систему на інші комп'ютери без необхідності складних процедур міграції або налаштування серверної інфраструктури. Ця особливість особливо цінна для організацій з обмеженими IT-ресурсами або для систем, які повинні бути розгорнуті в різних географічних локаціях з мінімальними зусиллями.

Більше того, відсутність необхідності в окремому серверному процесі значно знижує загальну складність системи, що в свою чергу підвищує її надійність та зменшує ймовірність виникнення збоїв, пов'язаних з проблемами мережевої комунікації або конфігурації сервера. Це особливо важливо для систем контролю доступу, де надійність та безперебійність роботи є критичними факторами. У випадку використання SQLite, додаток може працювати автономно, без залежності від зовнішніх сервісів або мережевих з'єднань, що робить систему більш стійкою до різноманітних збоїв та атак.

Другим ключовим фактором на користь вибору SQLite є її виняткова продуктивність та низьке споживання ресурсів. SQLite відома своєю високою швидкістю роботи та ефективністю, особливо в сценаріях з невеликим або середнім обсягом даних, що ідеально підходить для типової системи контролю доступу. Завдяки тому, що SQLite працює безпосередньо з файловою системою і не має проміжних шарів або процесів між додатком і даними, вона

забезпечує мінімальну затримку при виконанні запитів. Це особливо важливо в контексті систем контролю доступу, де швидкість відповіді системи на запит авторизації може бути критичним фактором, що впливає на зручність використання та загальну ефективність системи безпеки.

Низьке споживання ресурсів SQLite (рис. 3.1) також є значною перевагою, особливо для систем, які мають працювати на обладнанні з обмеженими обчислювальними можливостями або в умовах, де енергоефективність є важливим фактором. Це дозволяє розгортати систему контролю доступу на широкому спектрі пристроїв, від потужних серверів до вбудованих систем, одноплатних комп'ютерах (Raspberry Pi) або мобільних пристроїв, без значного впливу на їх продуктивність або час автономної роботи.

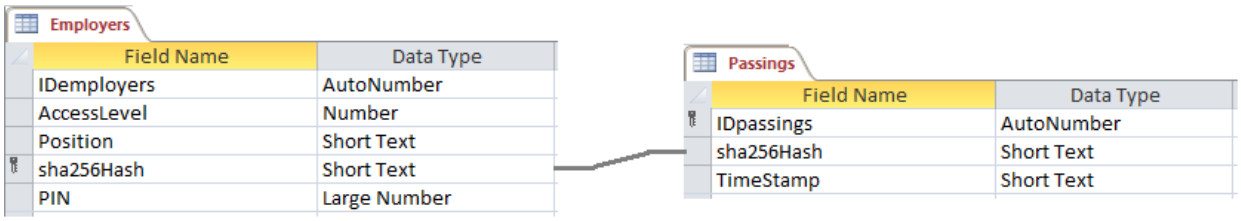

Рисунок 3.1 – Структура розробленої бази даних

Структура бази даних, розроблена для даної системи контролю доступу, складається з двох таблиць: "Employees" та "Passings". Ця структура є результатом ретельного аналізу та оптимізації, спрямованих на досягнення оптимального балансу між нормалізацією даних, продуктивністю системи та вимогами до безпеки та конфіденційності інформації. Кожна з цих таблиць відіграє ключову роль у функціонуванні системи та має свої унікальні особливості, які варто розглянути більш детально.

Таблиця "Employees" є центральним елементом бази даних, яка містить всю необхідну інформацію про співробітників та їх права доступу. Ключовими полями цієї таблиці є:

1. IDemployers: Це поле типу AutoNumber служить унікальним ідентифікатором кожного запису в таблиці. Використання автоінкрементного поля забезпечує унікальність кожного запису та спрощує процес додавання нових співробітників до системи.

2. AccessLevel: Поле типу Integer, яке визначає рівень доступу співробітника. У даній системі передбачено три рівні доступу: 1 - найнижчий рівень доступу, 2 - високий рівень доступу, 3 - найвищий рівень доступу. Така градація дозволяє гнучко налаштовувати права доступу для різних категорій співробітників, забезпечуючи принцип найменших привілеїв.

3. Position: Поле типу Text, яке зберігає інформацію про посаду співробітника. Це поле може бути використане для додаткової верифікації прав доступу або для генерації звітів та аналітики.

4. sha256Hash: Це поле типу Text є ключовим елементом системи безпеки та конфіденційності. Воно зберігає хеш-значення персональних даних співробітника (комбінаціz прізвища, імені та по батькові розділених пробілами), обчислене за допомогою алгоритму SHA-256. Використання цього поля як первинного ключа є інноваційним рішенням, яке забезпечує високий рівень захисту персональних даних. Замість зберігання ідентифікаційної інформації у відкритому вигляді, система оперує лише хеш-значеннями, що унеможливлює відновлення оригінальних даних навіть у випадку несанкціонованого доступу до бази даних.

5. PIN: Поле типу Integer, яке зберігає персональний ідентифікаційний номер співробітника. Цей PIN-код використовується як додатковий фактор аутентифікації, підвищуючи загальний рівень безпеки системи, і може використовуватися для забезпечення доступу в ручному режимі в разі виходу

з ладу апаратної частини системи контролю доступу чи виконавчого механізму.

Таблиця "Passings" є другим ключовим елементом бази даних, призначеним для детального логування всіх подій, пов'язаних з доступом до системи. Ця таблиця містить такі поля:

1. IDpassings: Поле типу AutoNumber, яке служить унікальним ідентифікатором кожного запису про спробу доступу. Це поле забезпечує хронологічну послідовність записів та спрощує процес аналізу історії доступу.

2. sha256Hash: Це поле типу Text є зовнішнім ключем, який пов'язує запис про спробу доступу з відповідним записом у таблиці "Employees". Використання хеш-значення замість прямого ідентифікатора співробітника забезпечує додатковий рівень анонімізації даних в логах.

3. TimeStamp: Поле типу DateTime, яке фіксує точний час спроби доступу. Ця інформація критично важлива для аудиту безпеки, розслідування інцидентів та аналізу патернів використання системи.

Така структура таблиці "Passings" дозволяє ефективно вести облік всіх входів та виходів, забезпечуючи при цьому високий рівень анонімізації даних. Це особливо важливо з точки зору відповідності вимогам законодавства про захист персональних даних, таким як GDPR в Європейському Союзі.

Важливо відзначити, що обрана структура бази даних оптимізована для роботи з REST API на основі Python/Flask. Flask, як легковагий та гнучкий фреймворк для створення веб-додатків, відмінно інтегрується з SQLite через ORM (Object-Relational Mapping) системи, такі як SQLAlchemy. Це дозволяє створити потужну абстракцію над базою даних, що значно спрощує розробку API-ендпойнтів та забезпечує гнучкість при роботі з даними.

Використання ORM надає ряд суттєвих переваг у контексті розробки REST API для системи контролю доступу:

1. Абстракція від SQL: Розробники можуть працювати з об'єктами Python замість прямих SQL-запитів, що підвищує читабельність коду та зменшує ймовірність помилок, пов'язаних з неправильно сформованими запитами.

2. Безпека: ORM автоматично екранує вхідні дані, що значно знижує ризик SQL-ін'єкцій та інших атак, пов'язаних з маніпуляцією вхідними даними.

3. Портативність: У разі необхідності міграції на іншу СУБД у майбутньому, більшість коду може залишитися незмінною, оскільки ORM абстрагує специфіку конкретної бази даних.

4. Продуктивність: Сучасні ORM, такі як SQLAlchemy, мають вбудовані механізми оптимізації запитів, що може покращити загальну продуктивність системи.

У контексті REST API, операції CRUD (Create, Read, Update, Delete) можуть бути легко реалізовані через відповідні HTTP-методи:

- POST запити можуть використовуватися для створення нових записів (наприклад, додавання нового співробітника).

- GET запити - для отримання інформації (наприклад, перевірка прав доступу або генерація звітів).

- PUT або PATCH запити - для оновлення існуючих записів (наприклад, зміна рівня доступу співробітника).

- DELETE запити - для видалення записів (наприклад, при звільненні співробітника).

Така структура API робить його інтуїтивно зрозумілим для розробників клієнтських додатків та відповідає принципам REST архітектури.

Використання SQLite в поєднанні з Python/Flask також забезпечує високу продуктивність при обробці паралельних запитів, що є критично важливим для системи контролю доступу, яка може обслуговувати велику

кількість користувачів одночасно. Хоча SQLite за замовчуванням підтримує лише одночасний запис (що може бути обмеженням для систем з високим навантаженням на запис), це рідко є проблемою для типових сценаріїв використання систем контролю доступу, де кількість операцій запису зазвичай невелика порівняно з операціями читання.

У разі необхідності масштабування системи для роботи з більшою кількістю одночасних користувачів або для обробки більшого обсягу даних, обрана структура бази даних дозволяє легко мігрувати на більш потужну СУБД, таку як PostgreSQL або MySQL, без суттєвих змін у структурі даних або логіці додатку. Це забезпечує довгострокову життєздатність та масштабованість рішення.

Особлива увага при проектуванні структури бази даних була приділена аспектам безпеки та захисту персональних даних. Використання хешування для зберігання ідентифікаційної інформації співробітників є важливим кроком у напрямку відповідності таким нормативним актам, як GDPR (General Data Protection Regulation) в Європейському Союзі або аналогічним законам в інших юрисдикціях. Це рішення забезпечує високий рівень захисту персональних даних співробітників, оскільки навіть у випадку несанкціонованого доступу до бази даних, зловмисник отримає лише хеш-значення, які неможливо зворотно перетворити на оригінальні дані.

Крім того, SQLite підтримує шифрування на рівні файлу бази даних, що додає додатковий шар захисту від несанкціонованого доступу. Це особливо важливо для систем контролю доступу, які часто працюють з конфіденційною інформацією та повинні відповідати суворим вимогам безпеки. Шифрування бази даних забезпечує захист даних не лише під час їх обробки, але й під час зберігання, що є критично важливим для запобігання витоку інформації у випадку фізичного доступу до сервера або пристрою, на якому розгорнута

система бо часто така система розгортається на машинах, що фізично розташовані на периферії і потенційно менш захищені і менш потужні.

Важливо відзначити, що обрана структура бази даних є легко масштабованою та адаптивною до майбутніх вимог. При необхідності додавання нових функцій, таких як підтримка різних типів аутентифікації (наприклад, біометрична аутентифікація або двофакторна аутентифікація), інтеграція з іншими системами безпеки (наприклад, системами відеоспостереження або контролю периметра) або впровадження більш складних механізмів авторизації, існуючу структуру бази даних можна легко розширити без порушення цілісності даних або необхідності повної реструктуризації. Наприклад, можна додати нові таблиці для зберігання біометричних даних або розширити існуючі таблиці додатковими полями для підтримки нових функцій, не змінюючи основну логіку роботи системи.

У контексті REST API на основі Python/Flask, обрана структура бази даних дозволяє ефективно реалізувати ряд ключових функцій системи контролю доступу:

1. Аутентифікація користувачів: Процес аутентифікації може бути реалізований через API-ендпойнт, який приймає хеш-значення персональних даних (наприклад, отримане від RFID-картки) та PIN-код. Сервер перевіряє відповідність цих даних записам у таблиці "Employees" і, у разі успішної аутентифікації, повертає токен доступу або інший ідентифікатор сесії. Використання хеш-значень замість відкритих персональних даних забезпечує високий рівень безпеки навіть при передачі даних по мережі.

2. Перевірка прав доступу: Після успішної аутентифікації, система може використовувати поле AccessLevel з таблиці "Employees" для визначення, чи має користувач право доступу до певного ресурсу або зони. Це може бути реалізовано через окремий API-ендпойнт або як частина процесу аутентифікації.

3. Логування подій: Кожна спроба доступу, успішна чи ні, може бути зареєстрована в таблиці "Passings" через відповідний API-ендпойнт. Це забезпечує повну прозорість всіх дій в системі та створює основу для подальшого аудиту безпеки та аналізу патернів використання системи.

4. Генерація звітів: API може надавати ендпойнти для отримання різноманітних звітів, наприклад, про активність користувачів за певний період, статистику успішних та невдалих спроб доступу, аналіз використання різних зон доступу тощо. Ці звіти можуть бути генеровані на основі даних з обох таблиць, "Employees" та "Passings".

5. Управління користувачами: REST API може надавати ендпойнти для додавання нових користувачів, оновлення інформації про існуючих користувачів (наприклад, зміна рівня доступу або PIN-коду) та видалення користувачів з системи. Всі ці операції будуть відображатися в таблиці "Employees".

6. Інтеграція з іншими системами: Обрана структура бази даних та REST API архітектура дозволяють легко інтегрувати систему контролю доступу з іншими корпоративними системами, такими як HR-системи, системи обліку робочого часу або системи безпеки. Наприклад, можна створити API-ендпойнти для синхронізації даних про співробітників з HR-системою або для передачі даних про доступ в систему обліку робочого часу.

Важливо відзначити, що у майбутньому при розробці REST API для роботи з даною структурою бази даних необхідно приділити особливу увагу питанням безпеки. Це включає використання HTTPS для шифрування всіх комунікацій між клієнтом та сервером, впровадження механізмів аутентифікації та авторизації для доступу до API (наприклад, використання JWT токенів), а також реалізацію захисту від поширених атак, таких як SQL-ін'єкції, CSRF та XSS.

Ще одним важливим аспектом буде оптимізація продуктивності API. Хоча SQLite відома своєю високою швидкістю роботи, при розробці API слід враховувати можливість високого навантаження, особливо в системах з великою кількістю користувачів.

Обрана структура бази даних також надає можливості для впровадження додаткових функцій безпеки та аудиту. Наприклад, можна реалізувати механізм відстеження змін (change tracking) для критично важливих полів у таблиці "Employees", зберігаючи історію змін у окремій таблиці. Це дозволить відслідковувати всі модифікації прав доступу чи інших важливих параметрів, що є важливим для забезпечення відповідності регуляторним вимогам та проведення внутрішніх аудитів безпеки.

Крім того, структура бази даних дозволяє легко реалізувати механізми резервного копіювання та відновлення даних. Оскільки вся база даних SQLite зберігається в єдиному файлі, процес резервного копіювання може бути реалізований шляхом простого копіювання цього файлу. Це значно спрощує процедури аварійного відновлення та забезпечує можливість швидкого відновлення системи у випадку збоїв або втрати даних.

Підсумовуючи, обрана структура бази даних SQLite в поєднанні з REST API на основі Python/Flask створює потужну, гнучку та безпечну основу для системи контролю доступу. Ця архітектура дозволяє ефективно зберігати та обробляти дані про співробітників та їх активність, забезпечуючи при цьому високий рівень захисту персональної інформації та відповідність сучасним стандартам безпеки та захисту даних..

3.5 Розробка REST API бекенду

Розробка REST API бекенду є ключовим етапом у створенні комп'ютеризованої системи контролю доступу з хешуванням персональних

даних. Цей компонент відіграє центральну роль у забезпеченні взаємодії між різними частинами системи, включаючи апаратну складову, базу даних та клієнтський інтерфейс. REST API сервер, реалізований на основі мови програмування Python та фреймворку Flask, представляє собою потужне та гнучке рішення, здатне ефективно обробляти запити, керувати даними та забезпечувати безпечну комунікацію між усіма елементами системи.

Вибір Python як основної мови програмування для реалізації REST API бекенду обумовлений кількома факторами. По-перше, Python відомий своєю простотою та читабельністю коду, що значно полегшує процес розробки та подальшого обслуговування системи. По-друге, Python має багату екосистему бібліотек та фреймворків, які дозволяють швидко та ефективно вирішувати різноманітні завдання. По-третє, Python добре підходить для роботи з даними та має потужні інструменти для обробки та аналізу інформації, що є критично важливим для системи контролю доступу.

Фреймворк Flask, обраний для створення REST API, є легковагим та гнучким інструментом, який ідеально підходить для розробки веб-додатків та API. Flask дозволяє швидко створювати масштабовані та ефективні веб-сервіси, при цьому надаючи розробнику повний контроль над архітектурою та компонентами системи. Використання Flask забезпечує високу продуктивність сервера, здатність обробляти велику кількість одночасних запитів та легку інтеграцію з іншими Python-бібліотеками та інструментами.

Одним з головних завдань REST API сервера є керування апаратною частиною системи, що здійснюється за допомогою бібліотеки pyserial. Ця бібліотека надає потужний та гнучкий інтерфейс для роботи з послідовними портами, дозволяючи встановлювати з'єднання, налаштовувати параметри комунікації та обмінюватися даними через COM-порт. Використання pyserial забезпечує надійну та ефективну комунікацію між програмним забезпеченням

та апаратною складовою системи, що є критично важливим для точної та своєчасної обробки запитів на доступ.

Процес взаємодії з RFID-зчитувачем починається з надсилання команди "r" (read) через COM-порт. Ця команда ініціює процес очікування на читання RFID-картки, переводячи апаратну частину в відповідний режим роботи. Реалізація цього механізму вимагає ретельного налаштування параметрів послідовного порту, включаючи швидкість передачі даних, біти даних, стоп-біти та контроль парності, щоб забезпечити стабільну та надійну комунікацію між сервером та RFID-зчитувачем.

Коли користувач підносить картку до зчитувача, система зчитує 64 символи з внутрішньої пам'яті RFID-картки. Ці символи представляють собою SHA-256 хеш у форматі HEX, який є унікальним ідентифікатором користувача. Вибір SHA-256 як алгоритму хешування обумовлений його високою криптографічною стійкістю та відсутністю відомих колізій, що забезпечує надійний захист персональних даних користувачів. Використання 64-символьного хешу дозволяє зберігати унікальний ідентифікатор користувача без необхідності зберігання його персональних даних у відкритому вигляді, що відповідає сучасним вимогам до захисту персональної інформації.

Отримавши 64-символьний хеш, REST API сервер виконує пошук відповідного запису у таблиці Employers бази даних SQLite. Для взаємодії з базою даних використовується ORM система SQLAlchemy, яка забезпечує зручний та безпечний спосіб роботи з даними. SQLAlchemy надає потужний набір інструментів для роботи з реляційними базами даних, дозволяючи абстрагуватися від деталей конкретної СУБД та працювати з даними на рівні об'єктів Python. Це не тільки спрощує процес розробки, але й підвищує безпеку системи, мінімізуючи ризики SQL-ін'єкцій та інших видів атак, пов'язаних з прямим доступом до бази даних.

Процес пошуку відповідного запису в базі даних вимагає ефективної організації даних та оптимізації запитів. Для підвищення продуктивності системи можуть бути застосовані різноманітні техніки оптимізації, такі як індексування полів, кешування частих запитів та використання підготовлених виразів. Крім того, важливо забезпечити коректну обробку ситуацій, коли відповідний запис не знайдено, або коли виникають помилки при роботі з базою даних.

Якщо співпадіння хешу не знайдено, сервер надсилає команду "d" (denied) назад до апаратної частини, відмовляючи у доступі, після чого знову надсилає команду "r" для переходу в режим очікування. Цей механізм забезпечує безпеку системи, запобігаючи несанкціонованому доступу. У випадку, коли співпадіння знайдено, сервер надсилає команду "g" (granted), дозволяючи доступ, і знову переходить в режим очікування командою "r". Важливо забезпечити надійну обробку помилок та логування всіх дій системи для можливості подальшого аудиту та аналізу подій безпеки.

REST API сервер також підтримує функціонал запису нового хешу на RFID-картку. Коли від інтерфейсу користувача надходить команда на запис хешу разом з 64-символьним значенням, сервер надсилає команду "w" (write), за якою слідує 64-символьний хеш, і завершує операцію командою "r" для повернення до режиму очікування. Цей функціонал є критично важливим для адміністрування системи, дозволяючи додавати нових користувачів або оновлювати дані існуючих. Реалізація цього механізму вимагає особливої уваги до безпеки, щоб запобігти несанкціонованому запису або модифікації даних на RFID-картках.

Окрім модуля комунікації з RFID через COM-port, REST API сервер реалізує чотири основні ендпоінти для виконання CRUD (Create, Read, Update, Delete) операцій над таблицею Employers бази даних SQLite. Ці ендпоінти доступні за наступними URL-адресами: localhost/create, localhost/read,

localhost/update та localhost/delete. Кожен з цих ендпоінтів приймає POST-запити з параметрами, що відповідають полям таблиці Employers: AccessLevel, Position, sha256Hash та PIN.

Ендпоінт localhost/create призначений для додавання нових записів до бази даних. При отриманні POST-запиту на цей ендпоінт, сервер перевіряє наявність всіх необхідних параметрів, виконує їх валідацію та створює новий запис у таблиці Employers. Важливо забезпечити перевірку унікальності sha256Hash для запобігання дублювання записів.

Ендпоінт localhost/read дозволяє отримувати інформацію про існуючі записи. Він може приймати параметри для фільтрації та сортування даних, що дозволяє гнучко налаштовувати вибірку інформації з бази даних. Результати запиту повертаються у форматі JSON, що забезпечує універсальність та зручність обробки даних на стороні клієнта.

Ендпоінт localhost/update призначений для оновлення існуючих записів. При отриманні POST-запиту сервер ідентифікує запис за допомогою sha256Hash і оновлює відповідні поля згідно з отриманими параметрами. Важливо забезпечити коректну обробку ситуацій, коли запис з вказаним хешем не знайдено.

Ендпоінт localhost/delete дозволяє видаляти записи з бази даних. Цей ендпоінт вимагає особливої уваги до безпеки, оскільки видалення даних є критичною операцією. Необхідно реалізувати механізми підтвердження видалення та, можливо, soft delete для можливості відновлення даних у разі помилкового видалення.

Використання ORM системи SQLAlchemy для реалізації CRUD операцій забезпечує ряд переваг. По-перше, це дозволяє абстрагуватися від специфіки конкретної СУБД, що полегшує можливе майбутнє перенесення системи на іншу базу даних. По-друге, SQLAlchemy надає потужні інструменти для оптимізації запитів та управління з'єднаннями з базою даних, що підвищує

продуктивність системи. По-третє, використання ORM значно знижує ризики, пов'язані з SQL-ін'єкціями, оскільки всі запити генеруються автоматично з використанням параметризованих виразів.

Важливим елементом REST API сервера є кореневий ендпоінт, доступний за адресою localhost/. Цей ендпоінт відповідає за повернення вмісту файлу index.html, який містить веб-інтерфейс користувача системи контролю доступу. Особливістю цього інтерфейсу є те, що він реалізований без використання Flask-інкапсуляцій, що дозволяє запускати його у будь-якому веб-браузері. Такий підхід забезпечує високу гнучкість та універсальність системи, дозволяючи користувачам працювати з нею через звичайний веб-браузер без необхідності встановлення додаткового програмного забезпечення, а функціонал реалізовано виключно через POST запити.

Веб-інтерфейс розроблений таким чином, щоб надсилати POST-запити до REST API сервера на його чотири CRUD ендпоінти, забезпечуючи повну функціональність управління системою через веб-браузер. Це включає можливість додавання нових користувачів, перегляду та редагування існуючих записів, а також видалення застарілих даних. Інтерфейс також надає функціонал для генерації звітів та аналізу активності користувачів системи.

Реалізація REST API бекенду на основі Python/Flask надає ряд переваг для системи контролю доступу. По-перше, це забезпечує високу гнучкість та розширюваність системи, дозволяючи легко додавати нові функції та модифікувати існуючі. Наприклад, можна легко інтегрувати додаткові методи аутентифікації, такі як двофакторна аутентифікація або біометричні дані, шляхом додавання відповідних ендпоінтів та модифікації логіки обробки запитів.

По-друге, використання Flask дозволяє створити легковагий та ефективний сервер, здатний обробляти велику кількість запитів. Flask надає інструменти для оптимізації продуктивності, такі як асинхронна обробка

запитів та кешування, що дозволяє системі ефективно працювати навіть при високому навантаженні.

По-третє, інтеграція з SQLAlchemy забезпечує надійну та безпечну роботу з базою даних, мінімізуючи ризики, пов'язані з SQL-ін'єкціями та іншими видами атак. SQLAlchemy надає потужні інструменти для управління схемою бази даних, що полегшує процес міграції та оновлення структури даних в процесі розвитку системи.

Важливо відзначити, що розробка REST API бекенду враховує аспекти безпеки, такі як валідація вхідних даних, захист від CSRF-атак та використання безпечних методів аутентифікації. Для забезпечення високого рівня безпеки можуть бути реалізовані додаткові механізми, такі як: Використання HTTPS для шифрування всіх комунікацій між клієнтом та сервером; Реалізація механізму токенів для аутентифікації та авторизації користувачів; Обмеження кількості запитів з одного IP-адресу для запобігання DoS-атакам; Логування всіх дій користувачів та спроб несанкціонованого доступу для можливості аудиту безпеки; Регулярне оновлення всіх використовуваних бібліотек та компонентів для усунення відомих вразливостей.

Реалізація цих механізмів безпеки вимагає ретельного планування та тестування, щоб забезпечити надійний захист системи без значного впливу на її продуктивність.

У додатках наведено лістинг програмного коду REST API сервера на основі Python/Flask, він дуже виразний і не потребує коментарів.

Цей код створює REST API сервер з наступними функціональними можливостями:

1. Керування апаратною частиною через COM-порт за допомогою бібліотеки `pyserial`.

2. Реалізація CRUD операцій для таблиці Employers через ендпоінти `/create`, `/read`, `/update`, `/delete`.

3. Використання SQLAlchemy ORM для роботи з SQLite базою даних.

4. Окремий потік для постійного читання RFID-карток та перевірки їх у базі даних.

5. Ендпоінт `/` для повернення файлу `index.html` з веб-інтерфейсом.

Також потрібно створити файл `index.html` в тій же директорії, що і цей скрипт. Цей файл містить WEB-інтерфейс користувача для взаємодії з REST API.

Також, на хост машині потрібно попередньовстановити залежності:

```
pip install flask flask-sqlalchemy pyserial jsonify send_file
```

Перед запуском скрипта треба переконатися, що вибрано `COM3` правильний номер COM-порта для хост системи.

3.6 Розробка WEB-інтерфейсу користувача

Розробка WEB-інтерфейсу користувача для комп'ютеризованої системи контролю доступу з хешуванням персональних даних є ключовим етапом у створенні зручного [18] та ефективного інструменту для управління системою. Програмний код наведено у додатках оскільки він занадто великий аби розмістити його тут, і код розділений на три файли: index.html, styles.css та script.js. Інтерфейс реалізовано з використанням сучасних веб-технологій: HTML для структурування контенту, CSS для стилізації та Vanilla JavaScript для забезпечення інтерактивності та взаємодії з REST API сервером.

Структура HTML документа розроблена з урахуванням семантики та доступності. Основний контейнер сторінки містить заголовок з назвою системи та навігаційне меню, яке дозволяє користувачеві перемикатися між різними режимами роботи: управління базою даних, форма для роботи з хеш-

значеннями та режим ручного управління доступом. Кожен з цих режимів представлений окремою секцією, яка відображається або приховується в залежності від вибору користувача.

Секція управління базою даних (рис. 3.2) містить форму для додавання нових користувачів, таблицю для відображення існуючих записів з можливістю редагування та видалення, а також елементи управління для генерації звітів.

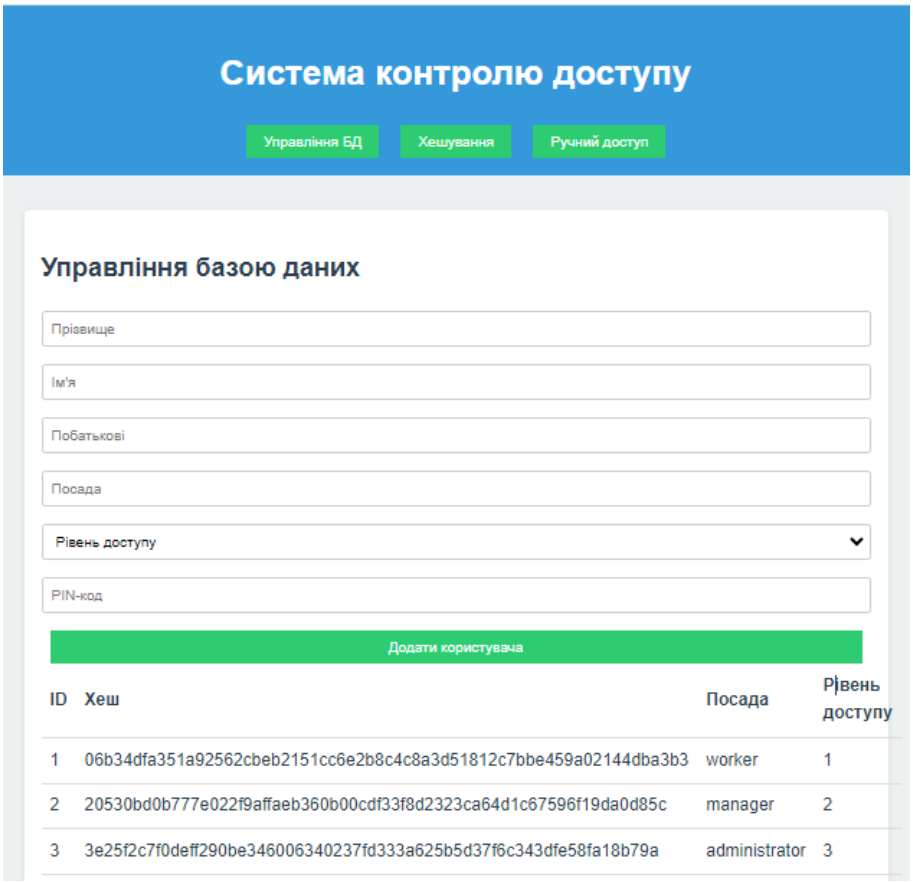

Рисунок 3.2 – WEB-інтерфейс системи. Секція «Управління базою даних»

Форма додавання користувача включає поля для введення імені, прізвища, посади, на основі яких вираховується хеш-значення SHA-256, рівня доступу та PIN-коду. Таблиця користувачів реалізована за допомогою HTML-

елемента <table> з динамічно генерованими рядками для кожного запису з бази даних.

Секція для роботи з хеш-значеннями (рис. 3.3) містить форму, де адміністратор може ввести персональні дані користувача (прізвище, ім'я, по батькові) для генерації хеш-значення SHA-256.

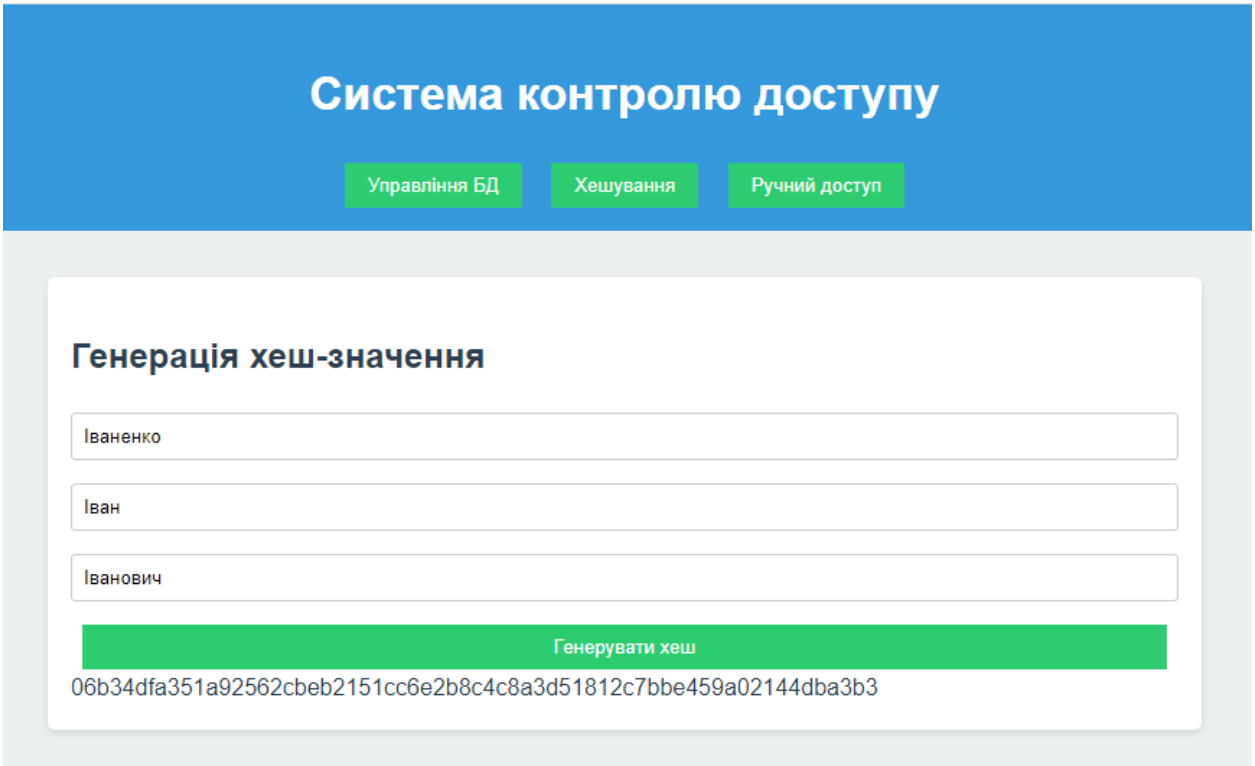

Рисунок 3.3 – WEB-інтерфейс системи. Секція «Генерація хеш-значення»

Результат хешування відображається в окремому полі, а також є можливість одразу скопіювати це значення в буфер обміну для подальшого використання.

Режим ручного управління доступом представлений формою, де можна ввести персональні дані та PIN-код для авторизації користувача в системі без використання RFID-картки (рис. 3.4). Ця форма також містить елементи для відображення результату авторизації та відкриття доступу.

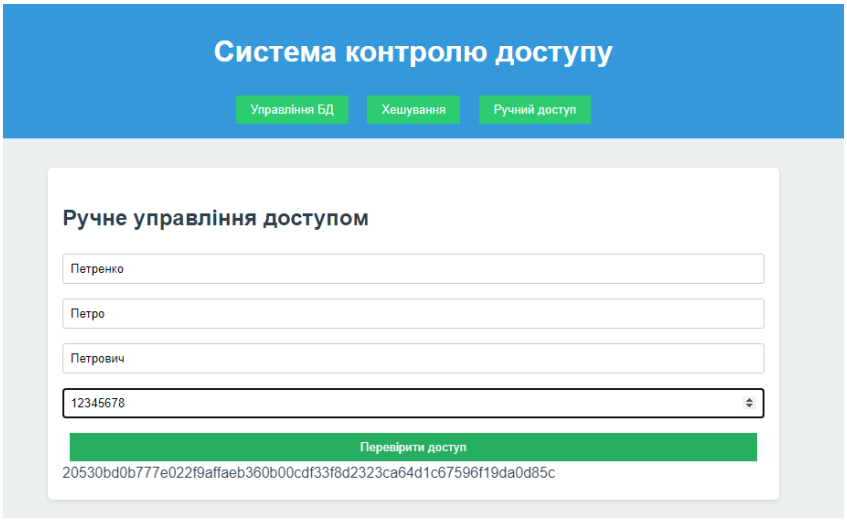

Рисунок 3.4 – WEB-інтерфейс системи. Секція «Ручне управління доступом»

Стилізація інтерфейсу виконана за допомогою CSS з використанням підходу mobile-first, що забезпечує адаптивність дизайну для різних розмірів екранів. Кольорова схема обрана з урахуванням психології кольору для забезпечення комфортного візуального сприйняття та підкреслення важливих елементів інтерфейсу. Використані CSS-змінні для легкого налаштування та зміни кольорової палітри. Анімації та переходи реалізовані за допомогою CSS-властивостей transition та animation для покращення користувацького досвіду при взаємодії з елементами інтерфейсу.

Інтерактивність та взаємодія з REST API сервером реалізовані за допомогою Vanilla JavaScript без використання сторонніх бібліотек чи фреймворків. Для кожної форми створені обробники подій, які перехоплюють відправку форми, збирають дані з полів вводу та формують відповідні POST-запити до сервера. Для відправки запитів використовується Fetch API, який дозволяє зручно працювати з асинхронними запитами.

Наприклад, при додаванні нового користувача, JavaScript-код перехоплює подію submit форми, збирає дані з полів вводу, формує об'єкт з даними користувача та відправляє POST-запит на відповідний ендпоінт REST API.

Для відображення результатів запитів та оновлення інтерфейсу використовуються функції, які динамічно змінюють DOM-структуру сторінки. Наприклад, функція updateUserTable() отримує актуальні дані про користувачів з сервера та оновлює вміст таблиці на сторінці.

Для покращення користувацького досвіду реалізовано відображення повідомлень про успішне виконання операцій або помилки. Ці повідомлення з'являються у вигляді спливаючих вікон і автоматично зникають через певний час.

Важливою частиною реалізації є обробка помилок та виключних ситуацій. Всі запити до сервера обгорнуті в конструкції try-catch, що дозволяє коректно обробляти мережеві помилки або помилки сервера і відображати відповідні повідомлення користувачеві.

Для забезпечення зручності використання на мобільних пристроях реалізовано адаптивний дизайн з використанням медіа-запитів CSS та JavaScript-коду для оптимізації відображення та функціональності на екранах різних розмірів.

В цілому, розроблений веб-інтерфейс користувача забезпечує зручний та ефективний інструмент для управління комп'ютеризованою системою контролю доступу, дозволяючи адміністраторам легко виконувати всі необхідні операції з управління користувачами та моніторингу активності в системі.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1. Особливості організації охорони праці при експлуатації комп'ютеризованої системи контролю доступу із хешуванням персональних даних

Організація охорони праці при експлуатації комп'ютеризованої системи контролю доступу із хешуванням персональних даних є важливим аспектом забезпечення безпеки та здоров'я працівників, а також захисту конфіденційної інформації. Комп'ютеризовані системи контролю доступу з хешуванням персональних даних є поширеним інструментом у різних галузях, таких як банківська справа, фінанси [19], охорона здоров'я, освіта [20] та ін. Вони дозволяють забезпечити безпеку та конфіденційність даних, а також контролювати доступ до них.

Однак, експлуатація таких систем може бути пов'язана з рядом ризиків для здоров'я та безпеки працівників. Наприклад, постійне використання комп'ютерів може призвести до фізичних та психологічних проблем, таких як біль у спині, головні болі, депресія та інші. Крім того, експлуатація таких систем може бути пов'язана з ризиком поширення інфекційних захворювань, через неадекватне використання особистого захисту та нестачу гігієнічних умов.

Також, експлуатація таких систем може бути пов'язана з ризиком поширення конфіденційної інформації, через несанкціонований доступ до неї. Це може призвести до серйозних наслідків для організації та її працівників, а також для індивідуальних осіб, чия інформація була поширена.

Організація охорони праці при експлуатації комп'ютеризованої системи контролю доступу із хешуванням персональних даних повинна бути спрямована на мінімізацію цих ризиків та забезпечення безпеки та здоров'я працівників. Для цього необхідні комплексні заходи, які включатимуть не тільки технічні та санітарно-гігієнічні вимоги, але й організаційні та психологічні аспекти.

У цьому розділі досліджено особливості організації охорони праці при експлуатації комп'ютеризованих систем контролю доступу із хешуванням персональних даних, а також фактори, які впливають на функціональний стан операторів таких систем. Цей процес регулюється низкою законодавчих та нормативних документів, серед яких ключову роль відіграють Закон України "Про охорону праці", Кодекс законів про працю України, Правила охорони праці під час експлуатації електронно-обчислювальних машин, а також стандарти ДСТУ 3008:2015, ДСТУ EN 62061:2019 та ДСТУ ISO/IEC 27001:2015.

Закон України "Про охорону праці" встановлює основні принципи та положення щодо реалізації конституційного права працівників на охорону їх життя і здоров'я у процесі трудової діяльності. Відповідно до ст. 13 цього закону, роботодавець зобов'язаний створити на робочому місці в кожному структурному підрозділі умови праці відповідно до нормативно-правових актів, а також забезпечити додержання вимог законодавства щодо прав працівників у галузі охорони праці. При експлуатації комп'ютеризованої системи контролю доступу це передбачає забезпечення належного технічного стану обладнання, проведення інструктажів з охорони праці та навчання персоналу безпечним методам роботи.

Кодекс законів про працю України регулює трудові відносини між працівниками та роботодавцями і встановлює їхні взаємні права та обов'язки. Згідно зі ст. 153 КЗпП, на всіх підприємствах, в установах, організаціях створюються безпечні і нешкідливі умови праці. Забезпечення безпечних і

нешкідливих умов праці покладається на власника або уповноважений ним орган. При експлуатації комп'ютеризованої системи контролю доступу роботодавець повинен забезпечити відповідність робочих місць операторів ергономічним та санітарно-гігієнічним вимогам.

Правила охорони праці під час експлуатації електронно-обчислювальних машин (НПАОП 0.00-1.28-10) встановлюють вимоги безпеки та санітарно-гігієнічні вимоги до обладнання робочих місць користувачів комп'ютерів та працівників, які виконують обслуговування, ремонт та налагодження комп'ютерів, та роботи з застосування комп'ютерів, у тому числі до роботи з комп'ютеризованими системами контролю доступу. Відповідно до цих правил, роботодавець повинен забезпечити відповідність робочих місць операторів системи ергономічним вимогам (освітлення, мікроклімат, шум тощо), а також організувати проведення попереднього (під час прийняття на роботу) і періодичних (протягом трудової діяльності) медичних оглядів працівників.

Особливу увагу при експлуатації комп'ютеризованої системи контролю доступу слід приділити питанням інформаційної безпеки та захисту персональних даних. Така система повинна відповідати вимогам ДСТУ ISO/IEC 27001:2015 "Інформаційні технології. Методи захисту. Системи управління інформаційною безпекою. Вимоги". Цей стандарт установлює вимоги до розроблення, впровадження, функціювання, моніторингу, перегляду, підтримування та постійного вдосконалювання задокументованої системи управління інформаційною безпекою в контексті загальних ділових ризиків організації. Хешування персональних даних, яке застосовується в комп'ютеризованій системі контролю доступу, є одним із методів їх захисту, що дозволяє зберігати інформацію у вигляді хеш-кодів. Це значно ускладнює несанкціоноване використання персональних даних у разі їх витоку або злому системи.

Функціональна безпечність системи керування контролем доступу повинна відповідати вимогам ДСТУ EN 62061:2019 "Безпечність машин. Функціональна безпечність систем керування, пов'язаних з безпекою". Цей стандарт установлює вимоги та настанови щодо проєктування, інтеграції та валідації пов'язаних з безпекою електричних, електронних і програмовних електронних систем керування для машин. Застосування цього стандарту при розробці та експлуатації комп'ютеризованої системи контролю доступу дозволяє забезпечити необхідний рівень безпечності системи шляхом проведення оцінки ризиків, визначення необхідного рівня повноти безпечності, реалізації відповідних заходів щодо зниження ризиків до прийнятного рівня.

При оформленні документації з охорони праці, зокрема інструкцій, положень та звітів, слід керуватися вимогами ДСТУ 3008:2015 "Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання". Цей стандарт установлює загальні вимоги до структурних елементів і правил оформлювання звітів у сфері науки й техніки. Дотримання вимог цього стандарту забезпечує уніфікацію та стандартизацію документації з охорони праці, що сприяє її кращому розумінню та застосуванню на практиці.

Крім зазначених законодавчих та нормативних документів, при організації охорони праці при експлуатації комп'ютеризованої системи контролю доступу слід також враховувати вимоги інших нормативно-правових актів, зокрема:

- Порядок проведення атестації робочих місць за умовами праці (Постанова КМУ від 01.08.1992 № 442);

- Типове положення про службу охорони праці (НПАОП 0.00-4.35-04);

- Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці (НПАОП 0.00-4.12-05);

- Державні санітарні норми та правила «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу» (затв. наказом МОЗ України від 08.04.2014 № 248);

- Перелік робіт з підвищеною небезпекою (НПАОП 0.00-4.12-2005);

- Перелік робіт, де є потреба у професійному доборі (НПАОП 0.00-4.26-96) та інші.

Таким чином, організація охорони праці при експлуатації комп'ютеризованої системи контролю доступу із хешуванням персональних даних вимагає комплексного підходу з урахуванням вимог чинного законодавства та нормативних документів у сфері охорони праці, інформаційної безпеки та функціональної безпечності систем керування. Дотримання цих вимог дозволяє забезпечити безпечні та здорові умови праці для працівників, які експлуатують систему, а також захистити конфіденційну інформацію та персональні дані від несанкціонованого доступу та використання. При цьому важливу роль відіграє правильна організація робочих місць операторів системи, проведення навчання та інструктажів з охорони праці, застосування сучасних методів захисту інформації, зокрема хешування персональних даних, а також виконання вимог до функціональної безпечності системи керування. Належне документальне оформлення заходів з охорони праці згідно з вимогами ДСТУ 3008:2015 забезпечує їх ефективну реалізацію на практиці.

4.2 Фактори, що впливають на функціональний стан оператора комп’ютера

Функціональний стан оператора комп'ютера є важливим фактором, який впливає на ефективність та безпечність його роботи. На нього можуть

впливати різноманітні чинники, зокрема фізичні, хімічні та психофізіологічні. Розглянемо кожну групу факторів детальніше.

Серед фізичних факторів, що впливають на функціональний стан оператора комп'ютера, слід виділити електромагнітні поля, іонізуюче випромінювання, шум, мікроклімат та освітлення робочого місця.

Електромагнітні поля, які генеруються комп'ютерним обладнанням, можуть чинити негативний вплив на здоров'я людини, зокрема викликати функціональні порушення нервової, ендокринної та серцево-судинної систем. Для мінімізації цього впливу необхідно забезпечити відповідність комп'ютерної техніки вимогам ДСанПіН 3.3.2.007-98 "Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин".

Іонізуюче випромінювання, яке може утворюватися при роботі старих моделей комп'ютерних моніторів на основі електронно-променевих трубок, здатне спричиняти функціональні зміни в організмі, зокрема в кришталику ока та сітківці. Сучасні рідкокристалічні монітори не створюють іонізуючого випромінювання, тому є безпечнішими для здоров'я операторів.

Шум, який виникає при роботі комп'ютерного обладнання (системних блоків, принтерів, сканерів тощо), може спричиняти зниження концентрації уваги, підвищення втомлюваності, зниження продуктивності праці. Для зменшення шумового навантаження на оператора необхідно забезпечити відповідність рівнів шуму вимогам ДСН 3.3.6.037-99 "Санітарні норми виробничого шуму, ультразвуку та інфразвуку" .

Параметри мікроклімату (температура, відносна вологість, швидкість руху повітря) та освітлення на робочому місці оператора комп'ютера повинні відповідати вимогам ДСанПіН 3.3.2.007-98 та ДБН В.2.5-28:2018 "Природне і штучне освітлення" . Відхилення цих параметрів від нормативних значень може призводити до погіршення самопочуття, зниження працездатності, розвитку професійних захворювань.

До хімічних факторів, які можуть впливати на функціональний стан оператора комп'ютера, належить забруднення повітря робочої зони шкідливими речовинами, що виділяються з комп'ютерного обладнання та офісних меблів. Це можуть бути фенол, формальдегід, полівінілхлорид, оксид вуглецю тощо. Для мінімізації впливу цих факторів необхідно забезпечити відповідність матеріалів і конструкції комп'ютерного обладнання та меблів встановленим санітарно-гігієнічним нормам, а також організувати ефективну систему вентиляції на робочому місці.

Психофізіологічні фактори, які можуть впливати на функціональний стан оператора комп'ютера, включають розумове перенапруження, монотонність праці, емоційні навантаження та ін. Робота з комп'ютером часто пов'язана з високою концентрацією уваги, аналізом великих обсягів інформації, прийняттям відповідальних рішень в умовах дефіциту часу. Це може призводити до розвитку психоемоційного стресу, синдрому "професійного вигорання".

Монотонність праці, яка характеризується виконанням одноманітних короткочасних операцій та значним навантаженням на окремі функціональні системи організму (зоровий аналізатор, м'язи кисті та передпліччя тощо), може викликати розвиток втоми, зниження продуктивності праці, погіршення здоров'я працівника.

Для профілактики несприятливого впливу психофізіологічних факторів на функціональний стан оператора комп'ютера необхідно забезпечити раціональну організацію режиму праці та відпочинку, зокрема передбачити наявність регламентованих перерв, під час яких рекомендується виконувати комплекс вправ для очей, рук, хребта тощо. Важливе значення має також проведення професійного відбору та психофізіологічної експертизи операторів.

Крім того, для підтримки високої працездатності та збереження здоров'я операторів комп'ютерів необхідно вживати заходів щодо загальної оптимізації

умов праці, зокрема забезпечувати комфортні параметри мікроклімату та освітлення, зниження рівня шуму, використання ергономічних меблів та обладнання, а також раціональне планування трудового процесу, яке передбачає чергування періодів роботи та відпочинку.

Таким чином, на функціональний стан оператора комп'ютера може впливати широкий спектр фізичних, хімічних та психофізіологічних факторів. Дотримання встановлених санітарно-гігієнічних норм та правил, раціональна організація робочого місця та трудового процесу, застосування профілактичних заходів дозволяють мінімізувати негативний вплив цих факторів та підтримувати високу працездатність і збереження здоров'я операторів.

Одним із важливих аспектів охорони праці операторів комп'ютерів є забезпечення безпечних умов праці при експлуатації комп'ютеризованих систем контролю доступу із хешуванням персональних даних. Такі системи можуть містити додаткові джерела електромагнітного випромінювання, шуму, вібрації, що потребує особливої уваги при організації робочих місць.

Крім того, робота з персональними даними накладає додаткові вимоги щодо захисту інформації та конфіденційності. Оператори повинні бути ознайомлені з правилами роботи з персональними даними, пройти відповідне навчання та інструктажі. Необхідно забезпечити надійний захист персональних даних від несанкціонованого доступу, витоку, пошкодження тощо.

Важливим напрямком охорони праці операторів комп'ютерів є також проведення попереднього (при прийомі на роботу) та періодичних (протягом трудової діяльності) медичних оглядів. Це дозволяє своєчасно виявляти ознаки професійних захворювань, надавати необхідну медичну допомогу та корегувати умови праці.

Не менш важливим є питання навчання та інструктажів з охорони праці. Оператори комп'ютерів повинні знати основні вимоги безпеки при роботі з

комп'ютерною технікою, вміти надавати першу медичну допомогу при нещасних випадках, бути ознайомлені з планами евакуації та діями в надзвичайних ситуаціях.

Крім того, для забезпечення високого рівня охорони праці операторів комп'ютерів необхідно проводити постійний контроль за дотриманням вимог безпеки, здійснювати аналіз виробничого травматизму та професійної захворюваності, розробляти та впроваджувати заходи щодо їх профілактики.

Таким чином, охорона праці операторів комп'ютерів є комплексною системою організаційних, технічних, санітарно-гігієнічних, лікувально-профілактичних заходів, спрямованих на збереження життя, здоров'я та працездатності працівників у процесі трудової діяльності. Її ефективна реалізація дозволяє мінімізувати негативний вплив шкідливих та небезпечних виробничих факторів, забезпечити безпечні умови праці та збереження здоров'я операторів комп'ютерів.

ВИСНОВКИ

Комп'ютеризована система контролю доступу з хешуванням персональних даних є складним проектом, який складається з двох основних компонентів: апаратної частини та програмної частини. Апаратна частина включає в себе плату Arduino UNO, модуль RFID-зчитувача MFRC522 та сервопривід для керування механізмом відкриття дверей. Ці компоненти обрані за їхню високу продуктивність, низьке енергоспоживання та сумісність з різноманітними протоколами RFID.

Програмна частина системи складається з сервера на основі Python, який забезпечує взаємодію між апаратною частиною та базою даних SQLite. Сервер обробляє дані, отримані від RFID-зчитувача, та зберігає їх у базі даних з використанням хеш-функцій для захисту персональних даних. Це забезпечує високу безпеку та цілісність даних, оскільки навіть у випадку несанкціонованого доступу до бази даних, зловмисник не зможе отримати доступ до реальних персональних даних користувачів.

Система розроблена з урахуванням принципів модульності та масштабованості, що дозволяє легко додавати нові функції та компоненти без необхідності повної переробки архітектури системи. Це особливо важливо для організацій, які постійно розвиваються та потребують системи, яка може адаптуватися до їхніх змінних потреб.

Окрім того, система розроблена з урахуванням вимог законодавства щодо захисту персональних даних, таких як GDPR в Європейському Союзі. Це означає, що система забезпечує збір та обробку тільки необхідних персональних даних, надає можливість користувачам отримувати доступ до своїх даних та видаляти їх за запитом, а також забезпечує прозорість обробки персональних даних відповідно до законодавчих вимог.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Benantar, Messaoud. Access control systems: security, identity management and trust models. Springer Science & Business Media, 2005.

2. Farooq, Umar, Mahmood ul Hasan, Muhammad Amar, Athar Hanif, and Muhammad Usman Asad. "RFID based security and access control system." International Journal of Engineering and Technology 6, no. 4 (2014): 309.

3. Kitsos, Paris, and Yan Zhang. "RFID security." Cham, Switzerland: Springer 233 (2008).

4. Ahson, Syed A., and Mohammad Ilyas. RFID handbook: applications, technology, security, and privacy. CRC press, 2017.

5. Van Tilborg, Henk CA, and Sushil Jajodia, eds. Encyclopedia of cryptography and security. Springer Science & Business Media, 2014.

6. Kayem, Anne VDM, Selim G. Akl, and Patrick Martin. Adaptive cryptographic access control. Vol. 48. Springer Science & Business Media, 2010.

7. Benantar, Messaoud. Access control systems: security, identity management and trust models. Springer Science & Business Media, 2005.

8. McElroy, Erin. "The work of landlord technology: The fictions of frictionless property management." Environment and Planning D: Society and Space (2024): 02637758241232758.

9. Bernard, Ray. Security Technology Convergence Insights. Elsevier, 2015.

10. Tréguer, F. (). Doing Action-Research on Algorithmic Urban Policing: IA-Powered Surveillance, Elusive Democratic Oversight.

11. Paulus, S., Pohlmann, N., Reimer, H., & Wirtz, B. (2004). Biometric System Security. In Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe 2003 Conference (pp. 108-119). Vieweg+ Teubner Verlag.

12. Kunikowski, Wojciech, Ernest Czerwiński, Paweł Olejnik, and Jan Awrejcewicz. "An overview of ATmega AVR microcontrollers used in scientific

research and industrial applications." Pomiary Automatyka Robotyka 19, no. 1 (2015): 15-19.

13. MFRC522 Standard performance MIFARE and NTAG frontend. Режим доступу: https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf

14. Led Tricolor RGB 5 mm - Common Anode. Режим доступу: https://www.electronicoscaldas.com/datasheet/LED5D-RGB-CA.pdf

15. Velychko, Diana, Halyna Osukhivska, Yuri Palaniza, Nadiia Lutsyk, and Łukasz Sobaszek. "Artificial Intelligence Based Emergency Identification Computer System." Advances in Science and Technology. Research Journal 18, no. 2: 296-304.

16. MG996R High Torque Metal Gear Dual Ball Bearing Servo. Режим доступу: https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf

17. Марків, В. А., Г.М. Осухівська, Ю.З. Лещишин, and Андрій Мирославович Луцків. "Комп'ютерна система аутентифікації осіб." Матеріали II наукової конференції Тернопільського національного технічного університету імені Івана Пулюя, 2017. С. 90-91.

18. Осухівська Г.М., Тиш Є.В., Луцик Н.С., Паламар А.М. Методичні вказівки до виконання кваліфікаційних робіт здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 123 «Комп'ютерна інженерія» усіх форм навчання. Тернопіль, ТНТУ. 2022. 28 с.

19. Горкуненко, А. Б., С. А. Лупенко, Г. М. Осухівська. Порівняльний аналіз математичних моделей циклічних економічних процесів в інформаційних системах підтримки прийняття економічних рішень. Науковий вісник НЛТУ України 22, no. 5.2012. С. 345-351.

20. Білостоцький, Т., Осухівська, Г.М., . Математичне моделювання передачі даних в комп'ютерних мережах. Матеріали II науково-технічної конференції "Інформаційні моделі, системи та технології". 2012. С.36-36.

Додаток А
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

«Затверджую»

завідувач кафедри КС

_____Осухівська Г.М.

"____"_____ 2024 р.

КОМП'ЮТЕРИЗОВАНА СИСТЕМА КОНТРОЛЮ ДОСТУПУ ІЗ ХЕШУВАННЯМ ПЕРСОНАЛЬНИХ ДАНИХ

ТЕХНІЧНЕ ЗАВДАННЯ

на __4__ листках

Вид робіт**:** Кваліфікаційна робота

На здобуття освітнього ступеня «Бакалавр»

Спеціальність 123 «Комп'ютерна інженерія»

«УЗГОДЖЕНО»	«ВИКОНАВЕЦЬ»
Керівник кваліфікаційної роботи	Студент групи СІс-42
_____к.т.н., ст. викл. Паляниця Ю.Б.	_____Наконечний В.В.
«___»_____2024 р.	«___»_____2024 р.

Тернопіль 2024

1. Повна назва та її умовне позначення.

Повна назва теми кваліфікаційної роботи «Комп'ютеризована система контролю доступу із хешуванням персональних даних».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.323.00.00

2. Виконавець

Студент групи СІс-42 факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Наконечний Віталій Володимировичё.

3. Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№4/7-408 від 24.04.2024 р.)

4. Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 01.02.2024 р.

Плановий термін завершення виконання кваліфікаційної роботи – 27.06.2024 р.

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ГОСТ, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником

роботи. Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

5. Призначення і цілі створення системи

Система що розробляється призначена для контролю доступу на основі RFID безконтактної технології із хешуванням персональних даних.

6. Мета створення системи

Метою кваліфікаційної роботи є розробка система що контролює доступ до об'єктів з обмеженим або режимним доступу на основі RFID безконтактної технології із хешуванням персональних даних для повного виключення їх можливого витоку.

7. До складу виробу повинні входити:

а) Плата Arduino UNO R3;

б) RFID модуль MFRC522;

в) Резистори номіналом 330 Ом;

г) Сервопривод MG996R.

8. Конструктивні вимоги

Конструювання корпусу приладу контролю доступу із хешуванням персональних даних не передбачено. Для побудови системи використана сучасна компонентна база.

9. Техніко-економічні показники

Планова собівартість проєкту повинна становити не більше 2000 гривень.

10. Стадії та етапи проєктування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи

№	Назва етапів роботи	Термін виконання етапів роботи
1	Розробка та затвердження технічного завдання	01.02-09.02.2024
2	Аналіз технічного завдання та обґрунтування можливих рішень	05.02.-11.02.2024
3	Розробка структурної схеми системи	01.06-03.06.2024
4	Розробка електричної принципової схеми, вибір елементної бази	02.06-10.06.2024
5	Розробка програмного забезпечення для проектованої системи	10.06-16.06.2024
6	Опрацювання питань розділу «Безпека життєдіяльності, основи охорони праці»	10.06-15.06.2024
7	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра	16.06-20.06.2024
8	Оформлення графічної частини	17.06-22.06.2024
9	Попередній захист кваліфікаційної роботи бакалавра	14.06.2024
10	Захист кваліфікаційної роботи бакалавра	24.06-28.06.2024

11. Під час виконання кваліфікаційної роботи у технічне завдання можуть вноситись зміни та доповнення.

Додаток Б

Перелік елементів

Поз. познач.	Найменування	Кіл.	Примітка
	Індикатор		
HL1	5218559F 5mm Discrete 4-lead RGB LED, Dialight	1	
	Мотор/привід		
M1	MG996R High Torque Metal Gear Servo, Terasic Technologies	1	
	Резистори		
R1-R3	Metal Film 330Ohm 1/8W ±5%, Vishay	3	
	Модулі/збірки		
U1	Arduino UNO R3, Microchip (Atmel)	1	
U2	MFRC522 RFID R/W Module, NXP	1	
XS1	MPX Connector 3-pin female, Neutrik	1	
XS2	USB1030-GF-P-B-B USB type B, GCT	1	

Змн.	Арк.	№ докум.	Підпис	Дата	КС КРБ 123.323.00.01 ПЕ3		
Розроб.		Наконечний В.В			Комп'ютеризована система контролю доступу із хешуванням персональних даних Перелік елементів	Літ.	Арк. 1 / Аркушів 3
Перевір.		ПаляницяЮ.Б.					
Н. Контр.		Тиш Є.В.				ТНТУ, каф. КС, гр. СІс-42	
Зав. каф.		Осухівська Г.М					
Реценз.		Ясній О.П.					

Додаток В

Лістинг програмного коду

Access_Control_System.ino:

```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);

Servo servo;
int ledPin = 13;

void setup() {
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  servo.attach(3);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  char command;
  if (Serial.available()) {
    command = Serial.read();
    switch (command) {
      case 'r':
        readRFID();
        break;
      case 'w':
        writeRFID();
        break;
      default:
        break;
    }
  }
}

void readRFID() {
  if (mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial()) {
    String cardData = "";
    for (byte i = 0; i < 16; i++) {
      cardData += String(mfrc522.uid.uidByte[i], HEX);
    }
    Serial.println(cardData);
    delay(5000); // Wait for response from computer
  }
}
```

```
void writeRFID() {
  // Receive 64-character hash code from computer
  String hash = "";
  while (hash.length() < 64) {
    if (Serial.available()) {
      hash += Serial.read();
    }
  }
  // Write hash to RFID card
  for (byte i = 0; i < 16; i++) {
    mfrc522.uid.uidByte[i] = strtol(hash.substring(i * 4, (i + 1)
* 4).c_str(), NULL, 16);
  }
  mfrc522.PICC_WriteCardSerial();
}

void accessGranted() {
  digitalWrite(ledPin, HIGH); // Green LED
  servo.write(90); // Unlock door
  delay(5000); // Wait for user to enter
  servo.write(0); // Lock door
  digitalWrite(ledPin, LOW); // Turn off LED
}
```

Access_Control_System.py:

```
from flask import Flask, request, jsonify, send_file
from flask_sqlalchemy import SQLAlchemy
import serial
import time
import os

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] =
'sqlite:///access_control.db'
db = SQLAlchemy(app)

# Налаштування серійного порту
ser = serial.Serial('COM3', 9600, timeout=1)  # '/dev/ttyUSB0'
порт відповідно до хост системи

class Employers(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    AccessLevel = db.Column(db.Integer)
    Position = db.Column(db.String(255))
    sha256Hash = db.Column(db.String(64), unique=True)
    PIN = db.Column(db.Integer)

def read_rfid():
    ser.write(b'r')
```

```
    time.sleep(0.1)
    if ser.in_waiting:
        return ser.read(64).decode()
    return None

def write_rfid(hash_value):
    ser.write(b'w' + hash_value.encode() + b'r')

@app.route('/')
def index():
    return send_file('index.html') # WEB-інтерфейс користувача

@app.route('/create', methods=['POST'])
def create():
    data = request.json
    new_employer = Employers(
        AccessLevel=data['AccessLevel'],
        Position=data['Position'],
        sha256Hash=data['sha256Hash'],
        PIN=data['PIN']
    )
    db.session.add(new_employer)
    db.session.commit()
    return jsonify({"message": "Employer created successfully"}),
201

@app.route('/read', methods=['POST'])
def read():
    data = request.json
    employer =
Employers.query.filter_by(sha256Hash=data['sha256Hash']).first()
    if employer:
        return jsonify({
            "AccessLevel": employer.AccessLevel,
            "Position": employer.Position,
            "sha256Hash": employer.sha256Hash,
            "PIN": employer.PIN
        })
    return jsonify({"message": "Employer not found"}), 404

@app.route('/update', methods=['POST'])
def update():
    data = request.json
    employer =
Employers.query.filter_by(sha256Hash=data['sha256Hash']).first()
    if employer:
        employer.AccessLevel = data.get('AccessLevel',
employer.AccessLevel)
        employer.Position = data.get('Position',
employer.Position)
        employer.PIN = data.get('PIN', employer.PIN)
        db.session.commit()
        return jsonify({"message": "Employer updated
successfully"})
```

```
    return jsonify({"message": "Employer not found"}), 404

@app.route('/delete', methods=['POST'])
def delete():
    data = request.json
    employer =
Employers.query.filter_by(sha256Hash=data['sha256Hash']).first()
    if employer:
        db.session.delete(employer)
        db.session.commit()
        return jsonify({"message": "Employer deleted
successfully"})
    return jsonify({"message": "Employer not found"}), 404

def rfid_loop():
    while True:
        rfid_data = read_rfid()
        if rfid_data:
            employer =
Employers.query.filter_by(sha256Hash=rfid_data).first()
            if employer:
                ser.write(b'gr')
            else:
                ser.write(b'dr')
        time.sleep(0.1)

if __name__ == '__main__':
    with app.app_context():
        db.create_all()

    # Запуск RFID loop у окремому потоці
    import threading
    rfid_thread = threading.Thread(target=rfid_loop)
    rfid_thread.start()

    app.run(debug=True, use_reloader=False)
```

index.html:

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Система контролю доступу з хешуванням персональних
даних</title>
     <link rel="icon" type="image/x-icon" href="favicon.ico">
    <link rel="stylesheet" href="styles.css">
</head>
<body>
```

```
    <header>
        <h1>Система контролю доступу</h1>
        <nav>
            <button id="showDbManagement">Управління БД</button>
            <button id="showHashForm">Хешування</button>
            <button id="showManualAccess">Ручний доступ</button>
        </nav>
    </header>

    <main>
        <section id="dbManagement" class="hidden">
            <h2>Управління базою даних</h2>
            <form id="addUserForm">
                <input type="text" name="lastName"
placeholder="Прізвище" required>
                <input type="text" name="firstName"
placeholder="Ім'я" required>
                <input type="text" name="position"
placeholder="Побатькові" required>
                <input type="text" name="position"
placeholder="Посада" required>
                <select name="accessLevel" required>
                    <option value="">Рівень доступу</option>
                    <option value="1">1 - Низький</option>
                    <option value="2">2 - Середній</option>
                    <option value="3">3 - Високий</option>
                </select>
                <input type="number" name="pin" placeholder="PIN-
код" required minlength="4" maxlength="8">
                <button type="submit">Додати користувача</button>
            </form>
            <table id="usersTable">
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Хеш</th>
                        <th>Посада</th>
                        <th>Рівень доступу</th>
                    </tr>
                </thead>
                <tbody id="usersTableRows">
                    <!-- ===== DEL ===== -->

    <tr><td>1</td><td>06b34dfa351a92562cbeb2151cc6e2b8c4c8a3d51812
c7bbe459a02144dba3b3</td><td>worker</td><td>1</td></tr><tr></tr>

    <tr><td>2</td><td>20530bd0b777e022f9affaeb360b00cdf33f8d2323ca
64d1c67596f19da0d85c</td><td>manager</td><td>2</td></tr><tr></tr>

    <tr><td>3</td><td>3e25f2c7f0deff290be346006340237fd333a625b5d3
7f6c343dfe58fa18b79a</td><td>administrator</td><td>3</td></tr><tr>
</tr>
                    <!-- ^^^^^ DEL ^^^^^ -->
                    </tbody>
```

```
            </table>
        </section>

        <section id="hashForm" class="hidden">
            <h2>Генерація хеш-значення</h2>
            <form id="generateHashForm">
                <input type="text" name="lastName"
placeholder="Прізвище" required>
                <input type="text" name="firstName"
placeholder="Ім'я" required>
                <input type="text" name="middleName"
placeholder="По батькові" required>
                <button type="submit">Генерувати хеш</button>
            </form>
            <div id="hashResult"></div>
        </section>

        <section id="manualAccess" class="hidden">
            <h2>Ручне управління доступом</h2>
            <form id="manualAccessForm">
                <input type="text" name="lastName"
placeholder="Прізвище" required>
                <input type="text" name="firstName"
placeholder="Ім'я" required>
                <input type="text" name="middleName"
placeholder="По батькові" required>
                <input type="number" name="pin" placeholder="PIN-
код" required minlength="4" maxlength="8">
                <button type="submit">Перевірити доступ</button>
            </form>
            <div id="accessResult"></div>
        </section>
    </main>

    <div id="messageContainer" class="hidden"></div>

    <script src="script.js" charset="utf-8"></script>
</body>
</html>
```

styles.css:

```
:root {
    --primary-color: #3498db;
    --secondary-color: #2ecc71;
    --background-color: #ecf0f1;
    --text-color: #2c3e50;
    --error-color: #e74c3c;
}

body {
```

```
    font-family: Arial, sans-serif;
    line-height: 1.6;
    color: var(--text-color);
    background-color: var(--background-color);
    margin: 0;
    padding: 0;
}

header {
    background-color: var(--primary-color);
    color: white;
    text-align: center;
    padding: 1rem;
}

nav {
    margin-top: 1rem;
}

button {
    background-color: var(--secondary-color);
    border: none;
    color: white;
    padding: 0.5rem 1rem;
    margin: 0 0.5rem;
    cursor: pointer;
    transition: background-color 0.3s;
}

button:hover {
    background-color: #27ae60;
}

main {
    max-width: 800px;
    margin: 2rem auto;
    padding: 0 1rem;
}

section {
    background-color: white;
    padding: 1rem;
    margin-bottom: 1rem;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

form {
    display: flex;
    flex-direction: column;
}

input, select {
    margin-bottom: 1rem;
```

```
    padding: 0.5rem;
    border: 1px solid #ccc;
    border-radius: 3px;
}

table {
    width: 100%;
    border-collapse: collapse;
}

th, td {
    text-align: left;
    padding: 0.5rem;
    border-bottom: 1px solid #ddd;
}

.hidden {
    display: none;
}

#messageContainer {
    position: fixed;
    top: 1rem;
    right: 1rem;
    padding: 1rem;
    border-radius: 5px;
    color: white;
    font-weight: bold;
}

.success {
    background-color: var(--secondary-color);
}

.error {
    background-color: var(--error-color);
}

@media (max-width: 600px) {
    nav {
        display: flex;
        flex-direction: column;
    }

    button {
        margin: 0.5rem 0;
    }
}
```

script.js:

```
// Глобальні змінні
const API_URL = 'http://localhost:5000/api'; // Замініть на вашу
адресу API
let authToken = localStorage.getItem('authToken');

// Функції для роботи з інтерфейсом
function showSection(sectionId) {
    document.querySelectorAll('section').forEach(section =>
section.classList.add('hidden'));
    document.getElementById(sectionId).classList.remove('hidden');
}

function displayMessage(message, type) {
    const messageContainer =
document.getElementById('messageContainer');
    messageContainer.textContent = message;
    messageContainer.className = type;
    messageContainer.classList.remove('hidden');
    setTimeout(() => {
        messageContainer.classList.add('hidden');
    }, 3000);
}

// Функції для роботи з API
async function makeApiRequest(endpoint, method, data = null) {
    const headers = {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${authToken}`
    };

    const options = {
        method: method,
        headers: headers,
        body: data ? JSON.stringify(data) : null
    };

    try {
        const response = await fetch(`${API_URL}${endpoint}`,
options);
        if (response.status === 401) {
            // Якщо токен недійсний, перенаправляємо на сторінку
входу
            window.location.href = 'login.html';
            return null;
        }
        return await response.json();
    } catch (error) {
        console.error('API request error:', error);
        displayMessage('Помилка з\'єднання з сервером', 'error');
        return null;
    }
}
```

```
// Функції для роботи з користувачами
async function addUser(userData) {
    const result = await makeApiRequest('/users', 'POST',
userData);
    if (result && result.success) {
        displayMessage('Користувача успішно додано', 'success');
        updateUserTable();
    } else {
        displayMessage('Помилка при додаванні користувача',
'error');
    }
}


async function updateUserTable() {
    const users = await makeApiRequest('/users', 'POST');
    if (users) {
        const tbody = document.querySelector('#usersTable tbody');
        tbody.innerHTML = '';
        users.forEach(user => {
            const row = tbody.insertRow();
            row.insertCell(0).textContent = user.lastName;
            row.insertCell(1).textContent = user.firstName;
            row.insertCell(2).textContent = user.position;
            row.insertCell(3).textContent = user.accessLevel;
            const actionsCell = row.insertCell(4);
            const deleteButton = document.createElement('button');
            deleteButton.textContent = 'Видалити';
            deleteButton.onclick = () => deleteUser(user.id);
            actionsCell.appendChild(deleteButton);
        });
    }
}


async function deleteUser(userId) {
    const result = await makeApiRequest(`/users/${userId}`,
'DELETE');
    if (result && result.success) {
        displayMessage('Користувача успішно видалено', 'success');
        updateUserTable();
    } else {
        displayMessage('Помилка при видаленні користувача',
'error');
    }
}


// Функції для ручного доступу
async function checkAccess(userData) {
    const result = await makeApiRequest('/access', 'POST',
userData);
    if (result) {
```

```
        document.getElementById('accessResult').textContent =
result.allowed ?
            'Доступ дозволено' : 'Доступ заборонено';
    } else {
        displayMessage('Помилка при перевірці доступу', 'error');
    }
}

// Обробники подій
document.getElementById('showDbManagement').addEventListener('clic
k', () => showSection('dbManagement'));
document.getElementById('showHashForm').addEventListener('click',
() => {showSection('hashForm'); runHashPolling_showHashForm();}) ;
document.getElementById('showManualAccess').addEventListener('clic
k', () => {showSection('manualAccess');
runHashPolling_showManualAccess();}) ;

document.getElementById('addUserForm').addEventListener('submit',
async (e) => {
    e.preventDefault();
    const formData = new FormData(e.target);
    const userData = Object.fromEntries(formData.entries());
    await addUser(userData);
});

document.getElementById('generateHashForm').addEventListener('subm
it', async (e) => {
    e.preventDefault();
    const formData = new FormData(e.target);
    const userData = Object.fromEntries(formData.entries());
    await generateHash(userData);
});

document.getElementById('manualAccessForm').addEventListener('subm
it', async (e) => {
    e.preventDefault();
    const formData = new FormData(e.target);
    const userData = Object.fromEntries(formData.entries());
    await checkAccess(userData);
});

// Ініціалізація
updateUserTable();
```