

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"  
УДК \_\_\_\_\_

«До захисту допущено»  
Завідувач кафедри

\_\_\_\_\_ О.В. Коваль  
(підпис) (ініціали, прізвище)

“ ” \_\_\_\_\_ 2019р.

## Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення  
за спеціалізацією Інженерія програмного забезпечення розподілених систем  
на тему Серверна частина веб-додатку інтернет-сервісу “Відкритий  
спортмайданчик з е-сервісами”

Виконав (-ла): студент (-ка) 6 курсу, групи ТВ-82мп  
Сербін Андрій Валерійович  
(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

Науковий керівник доц. к.т.н. Ковальчук А.М.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Рецензент \_\_\_\_\_  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_  
(підпис)

**Національний технічний університет України  
“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення

за спеціалізацією – Інженерія програмного забезпечення розподілених систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

Коваль О.В.

(прізвище, ініціали)

\_\_\_\_\_ (підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2019р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Сербіну Андрію Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації Серверна частина веб-додатку інтернет-сервісу “Відкритий спортмайданчик з е-сервісами”

Науковий керівник Ковальчук Артем Михайлович к.т.н. доц.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ року № \_\_\_\_\_

2. Строк подання студентом дисертації \_\_\_\_\_

3. Об'єкт дослідження Інформаційна система контролю та управління доступом

4. Предмет дослідження Програмні засоби та конфігурації розробки і хмарного розгортання інформаційних систем контролю та управління доступом

5. Перелік питань, які потрібно розробити Проведення огляду концепцій та основних понять систем контролю доступом, проведення порівняльного аналізу інструментів для створення веб-сервісів, дослідження підходів до побудови веб-сервісів, формулювання вимог до системи, розробка та тестування додатку для вирішення поставленої задачі, хмарне розгортання розробленого додатку, розроблення стартап-проекту

6. Орієнтований перелік ілюстративного матеріалу Архітектурна діаграма програмно-апаратного комплексу, архітектурна діаграма інтеграції з скуп, діаграма інтеграції сервісів в хмарному середовищі, схема бази даних, сваггер документація, діаграма класів, діаграма послідовностей

7. Перелік публікацій \_\_\_\_\_

1) Сербін А.В., Ковальчук А.М. Серверна частина веб-додатку інтернет сервісу “Відкритий спорт-майданчик з е-сервісами” // Збірник тез VI Всеукраїнської науково-практичної конференції молодих науковців. 16 травня 2019 р., м. Київ, Україна. – С. 106.

2) Сербін А.В., Ковальчук А.М. Розробка серверної частини для веб-додатку відкритий спорт-майданчик з е-сервісами // Матеріали XVII Міжнародної науково-практичної конференції молодих вчених та студентів. 23-26 квітня 2019 р., м. Київ, Україна. – С. 126.

8. Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 201\_\_ р.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів виконання магістерської дисертації     | Строки виконання етапів магістерської дисертації | Примітка |
|-------|---|--|----------|
| 1.    | Вивчення та аналіз задачі                           | 02.10.18 – 09.10.18                              |          |
| 2     | Розробка архітектури та загальної структури системи | 10.10.18 – 24.12.18                              |          |
| 3.    | Розробка окремих частин системи                     | 10.02.19 – 23.04.19                              |          |
| 4.    | Програмна реалізація системи                        | 24.04.19 – 27.10.19                              |          |
| 5     | Проходження переддипломної практики                 | 02.09.19 – 27.10.19                              |          |
| 6.    | Оформлення пояснювальної записки                    | 06.09.19 – 15.11.19                              |          |
| 7.    | Захист програмного продукту                         | 22.10.19 – 22.10.19                              |          |
| 8.    | Передзахист   | 21.11.19 – 21.11.19                              |          |
| 9.    | Захист  | 16.12.19 – 24.12.19                              |          |

Студент

\_\_\_\_\_  
( підпис )

Сербін А.В.

(прізвище та ініціали)

Науковий керівник

\_\_\_\_\_  
( підпис )

Ковальчук А.М.

(прізвище та ініціали)

## РЕФЕРАТ

**Структура та обсяг дипломної роботи.** Магістерська дисертація складається зі вступу, п'яти розділів, висновку, переліку посилань з 31 найменування, 4 додатки, і містить 15 рисунків, 24 таблиць. Повний обсяг магістерської дисертації складає 105 сторінок, з яких перелік посилань займає 2 сторінки, додатки — 23 сторінки.

**Актуальність теми.** В еру цифрових технологій активно впроваджуються системи, що використовують датчики, які ідентифікують співробітників набагато точніше і швидше, запобігають прохід сторонніх осіб і мінімізують людський фактор. Але ці системи розгорнуті на локальних серверах та внесення даних відбувається вручну через СУБД. В результаті чого додавання нових зчитувачів та контролерів чи додавання нового функціоналу є важким в реалізації, що негативно впливає на бізнес процеси закладу, тому дана тема є актуальною.

**Мета дослідження** полягає в розробці інформаційної системи контролю та управління доступом “Відкритий спортмайданчик”, яка буде гнучкою для змін чи додавання нового функціоналу, а також додасть автоматизацію в сферу контролю та управління доступом спортивного закладу.

Для досягнення поставленої задачі були сформульовані наступні **завдання дослідження**, що визначили логіку дослідження та його структуру:

- Дослідити можливі архітектурні підходи в побудові програмно-апаратного комплексу.
- Підібрати необхідне обладнання апаратної частини системи.
- Розрахувати економічне обґрунтування проекту.
- Розробити систему контролю і управління доступом на основі мікроконтролерів з використанням хмарних технологій.
- Розглянути варіанти подальшого поліпшення системи.

**Об'єктом дослідження** є інформаційна система контролю та управління

доступом.

**Предметом дослідження** є програмні засоби та конфігурації розробки і хмарного розгортання інформаційних систем контролю та управління доступом.

**Методи досліджень.** Проведення аналізу архітектур, хмарних середовищ, які потрібні для побудови ефективних програмно-апаратних комплексів, а також дослідження і вибір контролерів та зчитувачів для апаратної частини комплексу.

**Інноваційна новизна одержаних результатів.** Інноваційна новизна роботи полягає у створенні нових ефективних конфігурацій розробки та розгортання веб-сервісу, що може взаємодіяти з системами контролю доступом та є зручним для подальшої підтримки та розширення.

**Практичне значення одержаних результатів** полягає в розробці програмно-апаратного комплексу, з можливістю гнучкого масштабування, який надає віддалений доступ до системи з будь-якої точки світу за наявності підключення до мережі інтернет.

**Апробації результатів дисертації.** Основні положення роботи оприлюднено на:

1) VI Всеукраїнській науково-практичній конференції молодих науковців (м. Київ, 16 травня 2019 року).

2) XVII Міжнародній науково-практичній конференції молодих вчених та студентів (м. Київ, 23-26 квітня 2019 року).

#### **Публікації**

1) Сербін А.В., Ковальчук А.М. Серверна частина веб-додатку інтернет сервісу “Відкритий спорт-майданчик з е-сервісами” // Збірник тез VI Всеукраїнської науково-практичної конференції молодих науковців. 16 травня 2019 р., м. Київ, Україна. – С. 106.

2) Сербін А.В., Ковальчук А.М. Розробка серверної частини для веб-додатку відкритий спорт-майданчик з е-сервісами // Матеріали XVII Міжнародної науково-практичної конференції молодих вчених та студентів. 23-26 квітня 2019 р., м. Київ, Україна. – С. 126.

**Ключові слова.** Система контролю та управління доступом, хмарні технології, інформаційна система, веб-сервіс, мікроконтролер, зчитувач.

## РЕФЕРАТ

**Структура и объем дипломной работы.** Магистерская диссертация состоит из введения, пяти глав, заключения, списка ссылок с 31 наименованием, 4 приложения, и содержит 15 рисунков, 24 таблицы. Полный объем магистерской диссертации составляет 105 страниц, из которых перечень ссылок занимает 2 страницы, приложения — 23 страницы.

**Актуальность темы.** В эру цифровых технологий активно внедряются системы, использующие датчики, которые идентифицируют сотрудников гораздо точнее и быстрее, предотвращают проход посторонних лиц и минимизируют человеческий фактор. Но эти системы развернуты на локальных серверах и внесения данных происходит вручную через СУБД. В результате чего добавления новых считывателей и контроллеров или добавления нового функционала является тяжелым в реализации, негативно влияет на бизнес процессы заведения, поэтому данная проблема является актуальной.

**Цель исследования** заключается в разработке информационной системы контроля и управления доступом “Открытая спортплощадка”, которая будет гибкой для изменений или добавления нового функционала, а также придаст автоматизацию в сферу контроля и управления доступом спортивного заведения.

Для достижения поставленной задачи были сформулированы следующие **задачи исследования**, которые определили логику исследования и его структуру:

- Исследовать возможные архитектурные подходы в построении программно-аппаратного комплекса.
- Подобрать необходимое оборудование аппаратной части системы.
- Рассчитать экономическое обоснование проекта.
- Разработать систему контроля и управления доступом на основе микроконтроллеров с использованием облачных технологий.

- Рассмотреть варианты дальнейшего улучшения системы.

**Объектом исследования** является информационная система контроля и управления доступом.

**Предметом исследования** есть программные средства и конфигурации разработки и облачного развертывания информационных систем контроля и управления доступом.

**Методы исследований.** Проведение анализа архитектур, облачных сред, которые нужны для построения эффективных программно-аппаратных комплексов, а также исследования и выбор контроллеров и считывателей для аппаратной части комплекса.

**Инновационная новизна полученных результатов.** Инновационная новизна работы заключается в создании новых эффективных конфигураций разработки и развертывания веб-сервиса, который может взаимодействовать с системами контроля доступом и является удобным для дальнейшей поддержки и расширения.

**Практическое значение полученных результатов** заключается в разработке программно-аппаратного комплекса, с возможностью гибкого масштабирования, который предоставляет удаленный доступ к системе из любой точки мира при наличии подключения к сети интернет.

**Апробации результатов диссертации.** Основные положения работы обнародованы на:

1) VI Всеукраинской научно-практической конференции молодых ученых (г. Киев, 16 мая 2019).

2) XVIII Международной научно-практической конференции молодых ученых и студентов (г. Киев, 23-26 апреля 2019).

#### **Публикации.**

1) Сербин А.В., Ковальчук А.М. Серверная часть веб-приложения интернет сервиса “Открытый спорт-площадка с е-сервисами” // Сборник тезисов VI Всеукраинской научно-практической конференции молодых ученых. 16 мая 2019, г. Киев, Украина.. — С. 106.

2) Сербин А.В., Ковальчук А.М. Разработка серверной части для веб-приложения открытый спорт-площадка с e-сервисами // Материалы XVIII Международный научно-практической конференции молодых ученых и студентов. 23-26 апреля 2019, г. Киев, Украина.. — С. 126.

**Ключевые слова.** Система контроля и управления доступом, облачные технологии, информационная система, веб-сервис, микроконтроллер, считыватель.

# ABSTRACT

## **The structure and volume of the thesis.**

The master's thesis consists of an introduction, five sections, a conclusion, a list of links of 31 titles, 4 annexes, and contains 15 figures, 24 tables. The full volume of the master's thesis is 105 pages, of which the list of links occupies 2 pages, the annexes — 23 pages.

## **Actuality of theme.**

“Analog” methods of identification based on passports, passes, or other paper often do not meet current safety requirements. Also, these test methods significantly reduce throughput.

In the era of digital technology actively implemented systems using sensors that identify employees much more accurate and faster to prevent the passage of strangers and minimize the human factor. But these systems are deployed on local servers and entering data manually, via the database.

**The aim of the study** is to develop an information system access control “Outdoor playground” and adds automation in controlling and managing access to sports facilities. To provide ease of system expansion is necessary to analyze the problem that must be addressed.

To achieve this task, the following **research objectives** were formulated, which defined the logic of the study and its structure:

- To investigate the possible architectural approaches to building software and apartnoho complex.
- Pick the right equipment hardware system.
- Calculate economic assessment.
- Develop a system access control based microcontroller using cloud-based protocols.

**The object of study** is an information system access control.

**The subject of study** is a software development and configuration and deployment of cloud-based information systems access control.

**Methods.** Analysis architectures, cloud environments that are required to build hardware

and software, as well as research and selection of controllers and readers for the hardware industry.

**Innovative novelty of the results.** The innovative novelty of the work is to create new configurations effective development and deployment of Web services that can communicate with the access control system and is convenient for further support and expansion.

**The practical significance of the results** is to develop software and apartnoho complex, with the flexibility to scale, which provides remote access to the system from anywhere in the world when connected to the Internet.

**Approbation of dissertation results.**

The main provisions of work were reported and discussed at:

1) VI National scientific and practical conference of young scientists (Kyiv, May 16, 2019).

2) XVII International scientific conference of young scientists and students (Kyiv, 23 – 26 April 2019).

**Publications.**

1) Serbin A., Kovalchuk A. The server side web application online service “Outdoor playground with e-services” // Abstracts of the VI All-Ukrainian scientific conference of young scientists. May 16, 2019, Kyiv, Ukraine. — P. 106.

2) Serbin A., Kovalchuk A. Development of server side web application for outdoor sports ground with e-services // Proceedings of the XVII International scientific conference of young scientists and students. 23-26 April 2019, Kyiv, Ukraine. — P. 126.

**Keywords.**

Control and access management, cloud technologies, information systems, web services, microcontroller reader.

# ЗМІСТ

|   |    |
|---|----|
| СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....               | 14 |
| ВСТУП.....  | 16 |
| 1. СИСТЕМИ КОНТРОЛЮ І УПРАВЛІННЯ ДОСТУПОМ .....             | 17 |
| 1.1. Пристрої ідентифікації доступу .....                   | 18 |
| 1.2. Засоби управління доступом .....                       | 20 |
| 1.3. Виконавчі і керовані перегороджуючі пристрої.....      | 22 |
| 1.4. Аналіз існуючих рішень.....                            | 23 |
| 1.5. Постановка завдання роботи .....                       | 26 |
| 2. СТЕК ТЕХНОЛОГІЙ СИСТЕМ КОНТРОЛЮ І УПРАВЛІННЯ ДОСТУПОМ .. | 27 |
| 2.1. Клієнт-серверна архітектура.....                       | 27 |
| 2.1.1. Дворівнева архітектура .....                         | 28 |
| 2.1.2. Трирівнева архітектура .....                         | 29 |
| 2.1.3. Багаторівнева архітектура .....                      | 30 |
| 2.1.4. Representational State Transfer.....                 | 30 |
| 2.2. Хмарні обчислення .....                                | 31 |
| 2.2.1. Bluemix.....   | 32 |
| 2.2.2. Microsoft Azure.....                                 | 34 |
| 2.2.3. Amazon Web Services.....                             | 35 |
| 2.2.4. Force.com .....                                      | 36 |
| 2.2.5. Порівняльний аналіз хмарних провайдерів .....        | 37 |
| 2.3. Мова програмування .....                               | 39 |
| 2.4. Апаратна частина СКУД.....                             | 40 |
| 2.4.1. Система на базі Arduino .....                        | 41 |
| 2.4.2. Система на базі StopNet .....                        | 41 |
| 2.4.3. Система на базі Milestone та Biostar2 .....          | 44 |
| 2.4.4. Зчитувач Suprema Xpass S2 .....                      | 45 |

|  |    |
|--|----|
| 3. РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ .....               | 48 |
| 3.1. Реалізація програмної частини .....                       | 49 |
| 3.1.1. Реалізація бази даних .....                             | 50 |
| 3.1.2. Реалізація серверної програми .....                     | 52 |
| 3.2. Реалізація апаратної частини .....                        | 54 |
| 3.2.1. Підключення зчитувача до системи .....                  | 54 |
| 3.2.2. Інтеграція Biostar та VMS Milestone .....               | 55 |
| 4. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ .....                | 56 |
| 4.1. Використання програмної частини сервісу .....             | 56 |
| 4.2. Використання апаратної частини сервісу .....              | 58 |
| 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ .....                           | 59 |
| 5.1. Опис ідеї стартап-проекту .....                           | 59 |
| 5.2. Технологічний аудит ідеї проекту .....                    | 61 |
| 5.3. Аналіз ринкових можливостей запуску стартап-проекту ..... | 62 |
| 5.4. Розроблення ринкової стратегії проекту .....              | 71 |
| 5.5. Розробка маркетингової програми .....                     | 74 |
| 5.6. Економічне обґрунтування проекту .....                    | 77 |
| ВИСНОВКИ .....   | 79 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                               | 81 |
| Додаток А .....  | 83 |

## СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

СКУД — це комплекс технічних та програмних засобів безпеки, що здійснює регулювання входу / виходу та переміщень людей чи транспортних об'єктів на територіях, які знаходяться під охороною, для адміністративного моніторингу та попереджень несанкціонованого проникнення.

AWS — є дочірньою компанією Amazon.com, що надає платформу хмарних обчислень в оренду приватним особам, компаніям та урядам на основі платної підписки.

Web container — програма, що представляє собою сервер, який займається системною підтримкою сервлетів і забезпечує їх життєвий цикл відповідно до правил, визначеними в специфікаціях. Може працювати як повноцінний самостійний веб-сервер, бути постачальником сторінок для іншого веб-сервера, наприклад Apache, або інтегруватися в Java EE сервер додатків. Забезпечує обмін даними між сервлетом і клієнтами, бере на себе виконання таких функцій, як створення програмного середовища для функціонуючого сервлету, ідентифікацію та авторизацію клієнтів, організацію сесії для кожного з них.

HTTP (від англ. Hypertext Transfer Protocol) — протокол передачі гіпертекстових документів, що використовується в комп'ютерних мережах.

SDK (від англ. Software Development Kit) — набір із засобів розробки, утиліт і документації, який дозволяє програмістам створювати прикладні програми за визначеною технологією або для певної платформи (програмної або програмно-апаратної).

OAuth (скорочення від англ. Open Authorization) — це відкритий стандарт авторизації, який дозволяє користувачам відкривати доступ до своїх приватних даних (фотографій, відео, списки контактів), що зберігаються на одному сайті, іншому сайту, без необхідності вводу імені користувача або паролю.

Фреймворк (від англ. Framework, каркас, платформа) — інфраструктура програмних рішень, що полегшує розробку складних систем. Спрощено дану інфраструктуру можна вважати своєрідною комплексною бібліотекою.

API (від англ. Application Programming Interface) — прикладний програмний інтерфейс, набір визначень взаємодії різнотипного програмного забезпечення).

REST (від англ. Representational State Transfer, “передача репрезентативного стану”) — підхід до архітектури мережевих протоколів, які забезпечують доступ до інформаційних ресурсів.

JSON (від англ. JavaScript Object Notation) — це текстовий формат обміну даними між комп’ютерами.

JRE (від англ. — Java Runtime Environment) — мінімальна реалізація віртуальної машини, яка необхідна для виконання Java-додатків, без компілятора й інших засобів розробки.

## ВСТУП

Контроль відвідувачів організації, наприклад, у спортивних установах, захист від розкрадань, контроль порушення трудової дисципліни є одними з найсерйозніших проблем на більшості підприємств.

“Аналогові” методи ідентифікації, що ґрунтуються на паспортах, пропусках або інших паперових носіях часто не відповідають поточним вимогам безпеки. Також такі методи перевірки значно зменшують пропускну здатність.

В еру цифрових технологій активно впроваджуються системи, що використовують датчики, які ідентифікують співробітників набагато точніше і швидше, запобігають прохід сторонніх осіб і мінімізують людський фактор. Але ці системи розгорнуті на локальних серверах та внесення даних відбувається вручну через СУБД. В результаті чого додавання нових зчитувачів та контролерів чи додавання нового функціоналу є важким в реалізації, що негативно впливає на бізнес процеси закладу.

Тому, проблема полягає в тому, щоб розробити інформаційну систему контролю та управління доступом “Відкритого спортмайданчику”, яка буде гнучкою для змін чи додавання нового функціоналу, а також додасть автоматизацію в сферу контролю та управління доступом спортивного закладу.

Отже, актуальним є створення програмно-апаратного комплексу з можливістю гнучкого масштабування, який надасть сервіси керування системою через прикладний користувацький інтерфейс.

Розроблювана система допоможе аналізувати час перебування студентів на майданчику, обмежити доступ небажаних осіб, розмежувати права доступу до різних приміщень. Контролювати роботу системи буде можливо з будь-якої точки світу, маючи під рукою будь-який пристрій, здатний вийти в інтернет.

# 1. СИСТЕМИ КОНТРОЛЮ І УПРАВЛІННЯ ДОСТУПОМ

Системи контролю і управління доступом (СКУД) — це система технічних та програмних засобів безпеки, що здійснює регулювання входу і виходу та переміщень людей чи транспортних засобів на територіях з охороною, для адміністративного моніторингу та попереджень несанкціонованого проникнення[1].

СКУД мають широкий функціонал. Вони дозволяють контролювати:

- Доступ людей на об'єкт і в конкретні приміщення об'єкта.
- Час переміщення конкретних людей по об'єкту.
- Дії охорони під час чергування.

Крім цього, СКУД дозволяють вести табельний облік робочого часу кожного співробітника і фіксувати час приходу і відходу відвідувачів. Також СКУД можуть працювати спільно з системами охоронно-пожежної сигналізації і телевізійного відеоконтролю і здійснювати реєстрацію та видачу інформації про спроби несанкціонованого проникнення в приміщення, що охороняється.

Сучасні системи контролю та управління доступом включають в себе безліч компонентів. У цьому розділі будуть розглянуті основні компоненти СКУД: різні варіанти ідентифікаційних і зчитувальних пристроїв, пристрої контролю та управління доступу, виконавчі і перегороджувальні пристрої, а також засоби управління в складі апаратних пристроїв і програмних засобів[2].

Зазвичай системи контролю та управління доступом складаються з наступних основних компонентів:

- Пристрої ідентифікації: ідентифікатори і зчитувачі.
- Засоби управління доступом.
- Виконавчі пристрої і перегороджуючі керовані пристрої.

Також до складу різних СКУД можуть входити додаткові кошти — світлові або звукові сповіщувачі, джерела електроживлення, кнопки та інше. Окремі

пристрої СКУД можуть бути об'єднані в один блок (контролер зі зчитувачем) або бути відсутнім в сервері (персональний комп'ютер).

## 1.1. Пристрої ідентифікації доступу

Пристрої ідентифікації доступу включають в себе ідентифікатори і зчитувачі. Ідентифікатори представляють собою “унікальні ознаки суб'єкта або об'єкта доступу”. Ідентифікатором може бути будь-який код, біометричні характеристики або окрема ознака, код звуку.

Зчитувач — це “електронний пристрій, призначений для зчитування кодової інформації з ідентифікатора і перетворення її в стандартний формат, який передається для аналізу і ухвалення рішення в контролер”.

Таким чином, зчитувачі насамперед призначені для зчитування ідентифікаційної ознаки з ідентифікаторів, але також вони повинні забезпечувати введення коду і біометричної інформації і перетворювати отриману інформацію в потрібний формат для передачі на контролер (контролери) СКУД.

Зчитувачі між собою відрізняються за способом зчитування ідентифікаційних ознак. У зв'язку з цим виділяють зчитувачі з ручним введенням, контактні, безконтактні і комбіновані[3].

Зчитувачі, поряд з виконавчими пристроями, керованими пристроями встановлюються на контрольованих місцях, тобто безпосередньо там, де здійснюється контроль доступу. Наприклад, поряд з дверима, на турнікетах, кабінах проходу і так далі.

Ідентифікатори і зчитувачі відрізняються за видом використовуваних ідентифікаційних ознак (механічні, магнітні, оптичні, електронні контактні, радіочастотні, акустичні, біометричні і комбіновані).

Основні види відрізняються за способом запису, зберігання і зчитування кодової інформації, в зв'язку з чим забезпечують різний рівень надійності і різняться за ціною.

Існують наступні види кодів:

- Клавіатурний код, в такому випадку код вводиться через клавіатуру, кодовий перемикач і інший подібний пристрій.
- Фізичний код, код, записаний на матеріальному носії (карти, ключі, брелки).
- Біометричний код, являє собою ознаку або сукупність унікальних індивідуальних фізичних ознак людини.

Розглянемо найбільш поширені з існуючих на ринку види ідентифікаторів і відповідних до них зчитувачів.

Карти з штрих-кодом: на поверхню таких карт кольором нанесені смуги, ширина і відстань між якими складаються в кодову послідовність. Вартість таких карток низька — в районі 8 гривень, що дозволяє їх легко замінювати на нові при втраті. Вартість зчитувача так само низька — близько 2000 гривень.

Магнітні картки: на таких картках знаходяться магнітні смуги з кодом. Цей код може бути легко перепрограмований, що дозволяє швидко, дешево і без проблем при втраті старої картки переписати його на нову. Вартість карток і зчитувачів так само досить низька: картка коштує від 15 до 150 гривень, зчитувач від 1500 до 5000 гривень.

Віганд (Wiegand) карти: всередині картки знаходяться обрізки тонких металевих дротиків, розташованих в певному порядку. цей порядок являє собою кодову комбінацію. Розташування дротиків на картці фіксується спеціальним клеєм, після цього переорієнтація дротиків неможлива. Вартість картки і зчитувача як і раніше низька: картка коштує від 40 до 200 гривень, зчитувач від 5000 до 10000 гривень.

Проксимити (Proximity) карти: всередині картки розташована мікросхема (чіп) із записаною в ній інформацією. Проксимити картки діляться на активні і пасивні. У пасивних картках інформація записується один раз на весь час дії картки, а в активних існує можливість зміни інформації в мікросхемі. Вартість пасивних карток становить від 30 до 180 гривень, зчитувача від 250 до 40000 гривень. Вартість активних карток в 5 — 10 разів дорожче пасивних.

Електронні ключі (Touch Memory): ключі виконані у вигляді брелків. Всі необхідні дані записуються на укладену в них мікросхему. Вартість ключа і зчитувача середня: ключ коштує від 100 до 330 гривень, зчитувач в залежності від типу — 6000 до 25000 гривень.

Також можуть використовуватися ідентифікатори нижченаведених типів. Наприклад, ідентифікатори, що використовують цифрові клавіатури і ідентифікатори, які використовують біометричні характеристики. У ідентифікаторів першого типу носієм інформації є користувач, що набирає на клавіатурі замку особистий код (умовне число) і, якщо код вірний, то користувач отримує право доступу. Це простий і дешевий засіб контролю доступу, який, однак, легко обійти.

У ідентифікаторів другого типу інформацією є унікальні індивідуальні фізичні ознаки особистості: відбитки пальців, малюнок долоні, голос і так далі. Дуже надійні, проте застосовуються досить рідко через високу вартість (від 30000 до 150000 гривень), малої оперативності і великого обсягу машинної пам'яті, займаної одним “зліпком ключа”.

## **1.2. Засоби управління доступом**

За способом управління СКУД діляться на автономні, для управління одним або декількома пристроями без передачі інформації на центральний пристрій управління і контролю з боку оператора і на централізовані (мережеві) — для управління пристроями, що перегороджують обмін інформацією з центральним пультом та контролем і управлінням системою з боку центрального пристрою управління.

Засоби управління доступом включають в себе апаратні пристрої, тобто контролери та програмні засоби, тобто програмне забезпечення (ПЗ) СКУД.

Контролери є пристрої, які здійснюють обробку інформації від зчитувачів, які приймають рішення про доступ на підставі інформації, що надійшла і керують виконавчими пристроями і засобами оповіщення.

Існує ряд додаткових функцій, якими володіють деякі види контролерів. Наприклад, контролери можуть зберігати бази даних кодів користувачів, вибирати режим роботи, захищати від повторного використання картки, підключати до оповіщення охоронну сигналізацію і так далі.

В одній СКУД може знаходитися безліч контролерів, об'єднаних в єдину систему. Всі контролери однієї СКУД можуть підключатися або до ведучого контролера (майстер-контролера), або до комп'ютера, керуючого роботою всіх контролерів. Зазвичай провідний контролер відрізняється від інших тільки закладеною програмою. До нього може підключатися керуючий комп'ютер, принтер та інші периферійні пристрої[4].

Контролери бувають однофункціональними і багатфункціональними.

Однофункціональні контролери — це свого роду інтелектуальні аналоги кодового замка. Вони призначені для статичних даних та завжди працюють в автономному режимі.

Багатфункціональні контролери в свою чергу працюють і зі статичними, і з динамічними даними, а також не тільки управляють доступом, але і дозволяють стежити за станом виконавчих пристроїв. З використанням багатфункціональних контролерів можна створювати складні комплекси, інтегровані з іншими підсистемами безпеки.

Для зв'язку контролерів між собою в єдину мережу використовується стандартний інтерфейс RS 485. Зв'язок головного контролера з комп'ютером здійснюється за рахунок інтерфейсу RS 232. Багатфункціональні контролери працюють зазвичай в мережевому режимі: через централізований контроль і управління доступом.

Вартість контролерів в залежності від фірми виробника, номенклатури та комплексу поставки коливається від 5000 до 60000 гривень.

Для мережевих СКУД ПЗ забезпечує екранний інтерфейс для користувачів, занесення і зберігання ідентифікаційних кодів в пам'ять системи, завдання характеристик точок доступу, а також рівнів доступу для користувачів, встановлення тимчасових рамок доступу, протоколювання поточних і надзвичайних

подій, ведення і роботу баз даних (в тому числі їх резервне копіювання), реєстрація проходів через точки доступу до протоколів бази даних, виведення інформації про порушення і НС, а також управління пристроями, що на випадок аварійних і надзвичайних ситуацій.

Спеціальне програмне забезпечення дозволяє здійснювати автоматичний запис даних по всіх операціях входу / виходу. У будь-який момент доступні різноманітні відомості, наприклад, про місцезнаходження співробітників і відвідувачів. Поточний стан СКУД навіть відображається в графічній формі — виводиться план об'єкту, що охороняється, на якому стандартними значками вказуються зчитувачі, замки, технічні засоби охоронно-пожежної сигналізації, відеоконтролю і т.п., і система автоматично в реальному часі показує стан усіх завданих об'єктів контролю на плані.

Пристрої центрального управління, тобто персональні комп'ютери, призначені насамперед для програмування СКУД. Також вони використовуються для отримання інформації про користувачів системи, дату і час проходження користувачів через контрольні пристрої, спрацьовування засобів охоронно-пожежної сигналізації, спроб несанкціонованого проходження, аварійних ситуаціях, відеоконтролю і т.п.

Для роботи з СКУД може використовуватися будь-який персональний комп'ютер. Використовуючи спеціально розроблене для об'єкту, що охороняється програмне забезпечення, ПК здійснює загальне керівництво і програмування СКУД, а також збирає інформацію з контролерів, створює загальний банк даних і формує різні звіти і зведення.

### **1.3. Виконавчі і керовані перегороджуючі пристрої**

Виконавчі пристрої — це пристрої, які беруть команди управління з контролерів і призводять перегороджуючі пристрої у відкритий або закритий стан. Перегороджуючі пристрої (ПП) забезпечують фізичну перешкоду доступу (турнікети, шлюзи, прохідні кабінки, двері, ворота, і інші).

Перегороджуючі пристрої можуть по-різному обмежувати доступ. У зв'язку з цим виділяють наступні види ПП:

- З частковим перекриттям (турнікети, шлагбауми).
- З повним перекриттям (повно-ростові турнікети, спеціалізовані ворота).
- З суцільним перекриттям отвору (суцільні двері, ворота).
- З блокуванням об'єкта в отворі (шлюзи, кабінні прохідні).

У виконавчих пристроях застосовуються електромеханічні і електромагнітні замки, електромагнітні засувки (засувки та інше), а також механізми приводу.

Електромеханічні замки працюють за принципом переміщення закриваючих елементів (запорів, ригелів замків і т.п.) за допомогою включення на час їх пересування електромотора або електромагніту[5].

В електромагнітних замках відсутні рухомі механічні закриваючі елементи, тобто блокування пристроїв здійснюється за допомогою сил магнітного тяжіння від потужного магніту.

Електромагнітні засувки дозволяють змінювати своє положення вручну при відкритті або закритті в екстремальних умовах. Для повернення перегороджуючих пристроїв в закритий стан, вони дооборудуються спеціальними пристроями — доводчиками, без яких СКУД втрачають свою основну функцію — обмеження доступу, так як без них пристрій загородження може перебувати в будь-якому стані.

По виду виконавчого механізму доводчики поділяються на пружинні, пневматичні, гідравлічні і електромеханічні.

Доводчик не тільки гарантує закриття пристрою загородження, але і оберігає замок від механічних ударів, а при пожежі автоматично відчиняє двері і сприяє евакуації[6].

#### **1.4. Аналіз існуючих рішень**

У момент написання даної роботи, на українському ринку було представлено безліч систем контролю і управління, розглянемо деякі з них:

- PERCo-Web.

- ParsecNET Office.
- BOLID ОРІОН ПРО.
- Elsys Бастіон 2.

Система PERCo-Web має веб-інтерфейс. Інтерфейс системи містить 7 вкладок з підрозділами і безліччю кнопок і налаштувань, що робить його перевантаженим і складним для розуміння.

Вартість повної ліцензії на програмне забезпечення за поточним курсом становить приблизно 15 тисяч гривень. Доступний тестовий період на термін до 60 днів.

Для роботи цієї системи також необхідний сервер на ОС Windows Server версії не нижче 2008 R2 з встановленою СУБД Firebird 2.0. Загальна вартість обладнання, що включає в себе два контролера, чотири зчитувача карток і два електромеханічних замка, за поточним курсом становить приблизно 30 тисяч гривень. Також потрібно врахувати витрати на установку, кабелі та блоки живлення, міжблочні кабелі.

Основними плюсами системи є наявність веб-інтерфейсу і використання безкоштовної СУБД. З мінусів можна виділити високу вартість, закритий програмний код, незручний дизайн інтерфейсу, застарілу СУБД, підтримку тільки операційних систем сімейства Windows.

Система ParsecNET Office розроблена спеціально для об'єктів невеликого масштабу, що містять до 16 точок проходу. Компанія гарантує високу якість продукції, надає розширену гарантію на всю свою продукцію, а також безкоштовний підбір обладнання і ПЗ. Доступна безкоштовна версія ПЗ з обмеженим функціоналом.

Для функціонування системи, необхідний сервер на ОС Windows Server версії не нижче 2008 R2 з встановленою СУБД Microsoft SQL Server версії не нижче 2008 R2. Також, для контролю системи, потрібно обладнати робочі станції операторів. Мінімальний набір обладнання для системи складається з двох контролерів і чотирьох зчитувачів карт загальною вартістю 45650 гривень. Продажем та

підключенням електромеханічних замків компанія не займається. Загальна вартість ліцензій складає 60420 гривень.

Плюсами системи є висока якість продукції і безкоштовна підтримка. З мінусів можна виділити надмірно високу вартість, закритий програмний код, відсутність веб-інтерфейсу, підтримку тільки операційних систем сімейства Windows.

Система BOLID ОПІОН ПРО включає в себе крім системи контролю та управління доступом, ще й системи охоронно-пожежної сигналізації, відеоспостереження, автоматичного пожежогасіння, а також систему контролю та диспетчеризації об'єктів. Для функціонування СКУД потрібно купувати повну версію системи.

Програмне забезпечення можна встановити на будь-яку операційну систему сімейства Windows, починаю від версії XP. У ролі СУБД використовується Microsoft SQL Server Express 2012. Для роботи системи знадобиться купити два контролера доступу, чотири зчитувача карт і два електромагнітних замку загальною вартістю 15510 гривень.

Система зарекомендувала себе відносно низькою вартістю і можливістю підключити величезну кількість сторонніх датчиків. З мінусів можна виділити закритий програмний код, підтримку тільки операційних систем сімейства Windows.

Апаратно-програмний комплекс Бастіон-2 дозволяє інтегрувати в єдину систему відеоспостереження, охоронно-пожежні сигналізації, системи охорони периметра, системи охоронного освітлення, СКУД і системи управління технічними процесами. В якості СУБД використовується Oracle 11g.

Для функціонування СКУД досить купити серверну версію ПЗ і модуль для роботи СКУД. Загальна вартість становитиме 27400 гривень. Вартість двох контролерів і чотирьох лічильників становитиме 30000 гривень. Продажем та підключенням електромеханічних замків компанія не займається.

Плюсами системи можна відзначити наявність доступу через веб-інтерфейс і можливість доукомплектувати системи додатковими модулями. З мінусів можна

виділити високу вартість, закритий програмний код і підтримку тільки операційних систем сімейства Windows.

## **1.5. Постановка завдання роботи**

Метою дипломної роботи є створення програмно-апаратного комплексу з можливістю гнучкого масштабування, який надасть сервіси керування системою через прикладний користувацький інтерфейс.

Для досягнення вказаної вище мети, були поставлені наступні завдання:

- Дослідити можливі архітектурні підходи в побудові програмно-апаратного комплексу.
- Підібрати необхідне обладнання апаратної частини системи.
- Розрахувати економічне обґрунтування проекту.
- Розробити систему контролю і управління доступом на основі мікроконтролерів з використанням хмарних технологій.
- Розглянути варіанти подальшого поліпшення системи.

## 2. СТЕК ТЕХНОЛОГІЙ СИСТЕМ КОНТРОЛЮ І УПРАВЛІННЯ ДОСТУПОМ

Для розробки системного і прикладного програмного забезпечення в сучасних умовах характерні такі риси, як використання концепцій паралельних і розподілених обчислень, мережних технологій та засобів Інтернет.

### 2.1. Клієнт-серверна архітектура

Клієнт-сервер — обчислювальна або мережева архітектура, в якій завдання або мережеві навантаження розподілені між постачальниками послуг (серверами) і замовниками послуг (клієнтами). Архітектура клієнт-сервер — один із способів організації інформаційної системи[7].

Інформаційна система — сукупність даних (баз даних) і програм, що функціонують на обчислювальних засобах як єдине ціле для вирішення певних завдань. На рисунку 2.1 показані способи організації клієнт-серверної архітектури.

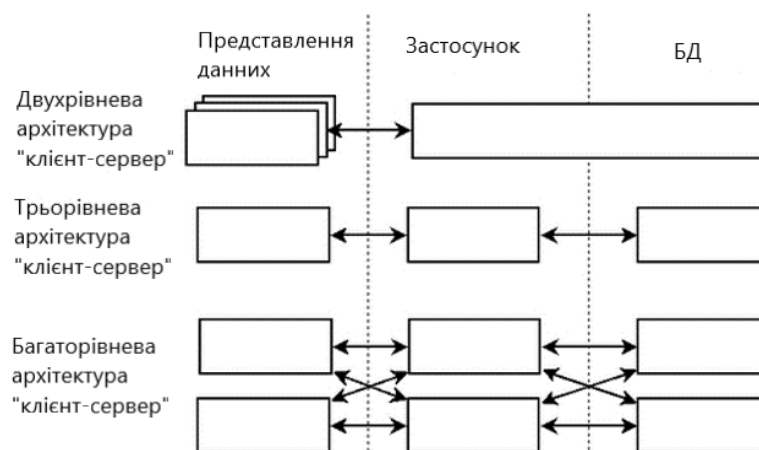


Рисунок 2.1 — Способи організації клієнт-серверної архітектури

У більшості інформаційних систем можна виділити основні групи функцій:

- Функції представлення даних.

- Функції, характерні для вирішення задач конкретної предметної області.
- Функції управління ресурсами.

В залежності від розподілу цих функцій між вузлами інформаційної системи, виділяють кілька різних варіантів реалізації архітектури клієнт-сервер.

### 2.1.1. Дворівнева архітектура

Найпростіший варіант реалізації архітектури. У ній присутні лише дві ланки — клієнт і сервер (рисунок 2.2).

Особливістю організації цього способу взаємодії є те, що сервер не звертається до жодних сторонніх сервісів, а використовує лише власні ресурси[8].

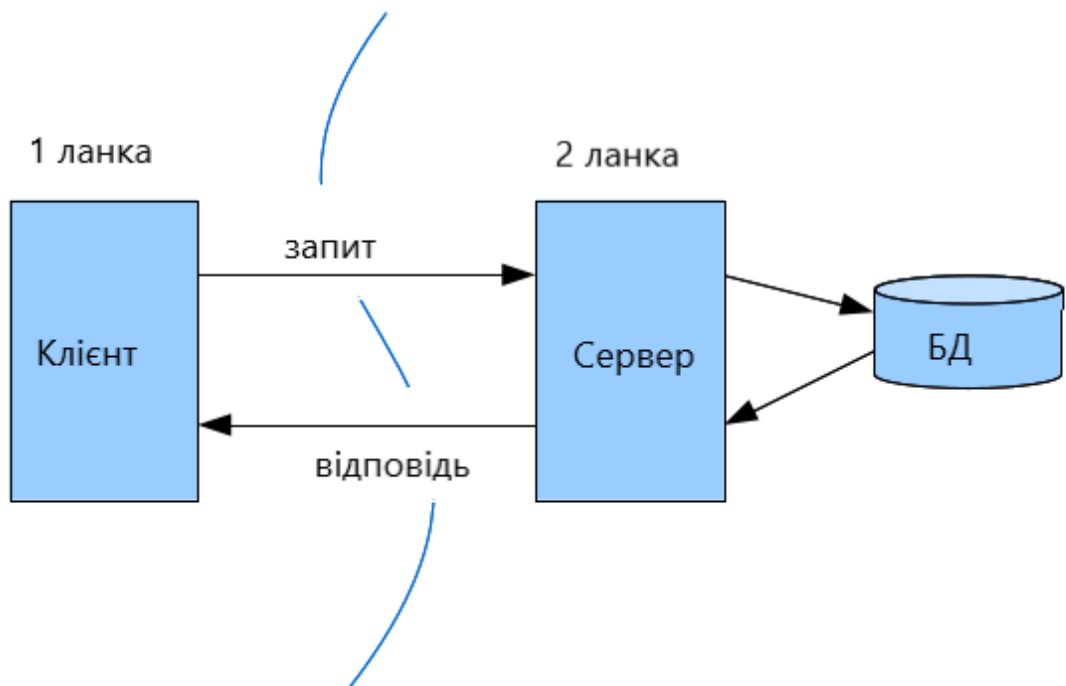


Рисунок 2.2 — Дворівнева клієнт-серверна архітектура.

Клієнтська частина містить рівень представлення даних, а на сервері знаходиться база даних разом з СУБД і прикладними програмами.

В рамках дворівневої архітектури визначені наступні основні моделі взаємодії клієнта і сервера:

- Сервер застосунків — віддалений застосунок.
- Файл-сервер — доступ до віддаленої бази даних і файлових ресурсів.

- Сервер терміналів — розподілене представлення даних.
- Сервер БД — віддалене уявлення даних.

### 2.1.2. Трирівнева архітектура

У зв'язку з розвитком розподілених обчислень, була виділена трирівнева архітектура, яка показана на рисунку 2.3.

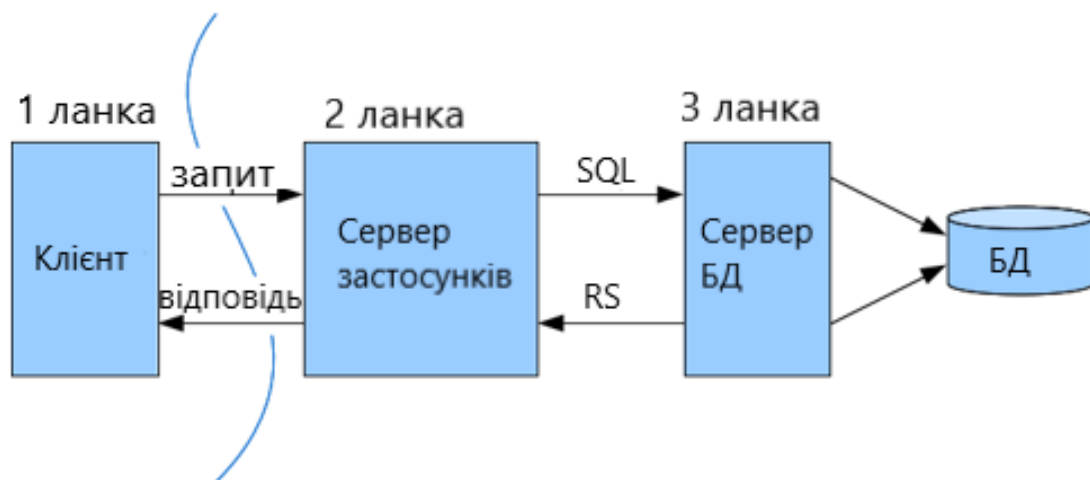


Рисунок 2.3 — Трирівнева клієнт-серверна архітектура

Вона реалізується на основі моделі сервера застосунків, де мережевий застосунок розділений на дві і більше частин, кожна з яких може виконуватися на окремій машині. Виділені частини додатка взаємодіють один з одним, обмінюючись повідомленнями в заздалегідь узгодженому форматі[9].

Як правило, третьою ланкою в триланковій архітектурі стає сервер застосунків, тобто компоненти розподіляються наступним чином:

- Представлення даних — на стороні клієнта.
- Прикладний компонент — на виділеному сервері застосунків.
- Управління ресурсами — на сервері БД, який представляє запитувані дані.
- Зв'язок між ресурсами — в окремому компоненті обміну повідомлень між ресурсами.

### 2.1.3. Багаторівнева архітектура

Трирівнева архітектура може бути розширена до багаторівневої шляхом виділення додаткових серверів, кожен з яких представлятиме власні сервіси і користуватися послугами інших серверів різного рівня[10]. Спосіб організації показаний на рисунку 2.4.

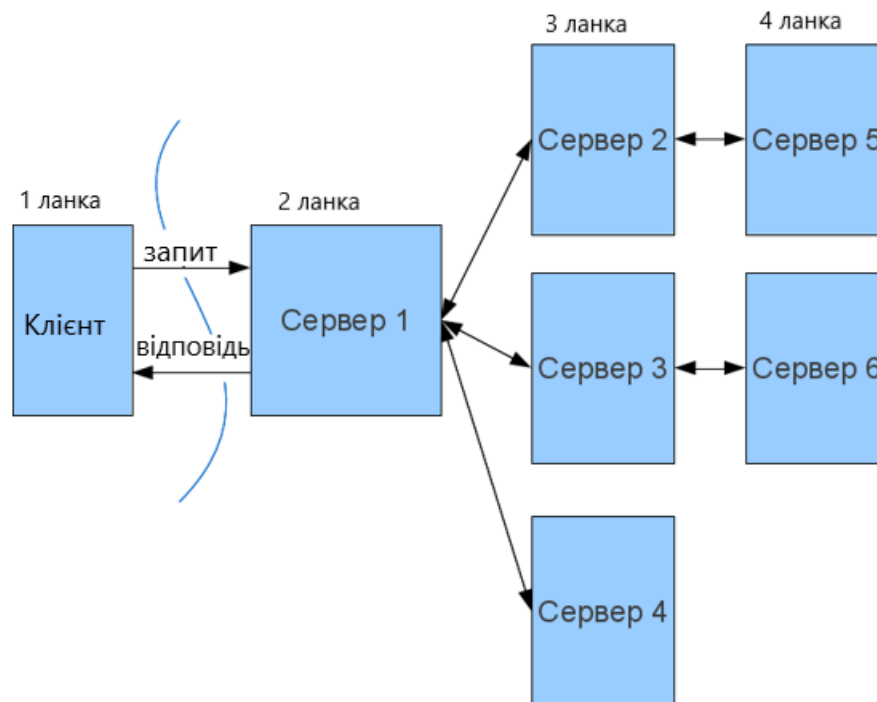


Рисунок 2.4 — Багаторівнева клієнт-серверна архітектура

Багаторівневі клієнт-серверні системи досить легко можна перевести на Web-технологію — для цього достатньо замінити клієнтську частину універсальним або спеціалізованим браузером, а сервер застосунків доповнити Web-сервером і невеликими програмами виклику процедур сервера[11]. Тому багаторівневу архітектуру було прийнято за основу в розроблюваному застосунку.

### 2.1.4. Representational State Transfer

Representational State Transfer (REST) — архітектурний стиль, який визначає набір обмежень та властивостей на основі HTTP[12]. Веб-сервіси, які відповідають

архітектурному стилю REST або RESTful web services, забезпечують сумісність між комп'ютерними системами в Інтернеті. REST-сумісні веб-сервери дозволяють запитуючим системам отримувати доступ до веб-ресурсів і керувати ними, використовуючи єдиний та попередньо визначений набір операцій, тому даний архітектурний стиль був обраний за основу в розроблюваному застосунку.

“Веб-ресурси” вперше були визначені в World Wide Web як документи або файли, визначені їх URL-адресами. Однак сьогодні вони мають набагато більш загальне і абстрактне визначення, яке охоплює всі речі або об'єкти, які можуть бути ідентифіковані, названі, адресовані або оброблені будь-яким способом в Інтернеті. У RESTful веб-службі запити, зроблені до URI ресурсу, викликають відповідь, яка може бути в HTML, XML, JSON або іншому форматі. Відповідь може підтвердити, що ресурс змінився. Відповідь також може надавати гіпертекстові посилання на інші пов'язані ресурси або колекції ресурсів[13]. Були використані такі HTTP методи:

- GET — запитує представлення зазначеного ресурсу. Запити, що використовують GET, повинні отримувати лише дані та не повинні мати інший ефект.
- POST — містить сутність в запиті, який стане новий підпорядкованим веб-ресурсом, ідентифікованим URI. Цими сутностями в основному є JSON об'єкти.
- PUT — запитує, щоб доданий об'єкт зберігався під наданим URI. Якщо URI відноситься до вже існуючого ресурсу, він змінюється; якщо URI не вказує на існуючий ресурс, то сервер може створити ресурс з цим URI.
- DELETE — видаляє вказаний ресурс.

## 2.2. Хмарні обчислення

Хмарні обчислення (Cloud computing) (англ. Cloud — хмара; computing — обчислення) — це модель надання повсюдного і зручного мережевого доступу в міру необхідності до загального потоку конфігурованих обчислювальних ресурсів (наприклад, мереж, серверів, систем зберігання, додатків і сервісів), які можуть бути

швидко надані і звільнені з мінімальними зусиллями з управління та необхідністю взаємодії з провайдером послуг (сервіс-провайдером)[14].

Хмарна модель підтримує високу доступність сервісів і описується п'ятьма основними характеристиками (essential characteristics), трьома сервісними моделями / моделями надання послуг (service models) і чотирма моделями розгортання (deployment models).

Програми запускаються і видають результати роботи в вікно стандартного веб-браузера на локальному ПК, при цьому всі додатки і їх дані, необхідні для роботи, знаходяться на віддаленому сервері в інтернеті. Комп'ютери, які здійснюють cloud computing, називаються “обчислювальною хмарою”. При цьому навантаження між комп'ютерами, що входять в “обчислювальну хмару” , розподіляється автоматично.

### 2.2.1. Bluemix

Bluemix — середовище розробки, створення і адміністрування застосунків в хмарі. Є реалізацією архітектури IBM Open Cloud Architecture на основі відкритого ПЗ Cloud Foundry, що працює за принципом PaaS (“платформа як послуга”).

Cloud Foundry — це ПЗ з відкритим вихідним кодом, яке представляє собою платформу, що дозволяє швидко створювати і розгортати застосунки в хмарі[15].

IBM Bluemix підтримує різні мови програмування:

- Java.
- PHP.
- Python.
- Ruby.
- Swift.
- GO.

Контейнери та платформи, що дозволяють розгортати і запускати різного роду програми:

- Node.js.

- Tomcat.
- ASP.NET 5.
- Xpages.

Середовище розробки IBM Bluemix має велику кількість сервісів, завдяки налаштуванням і комбінаціям яких можна створювати, розгортати і адмініструвати безліч застосунків (таблиця 2.1).

| <i>Найменування служби</i> | <i>Опис</i>   |
|----------------------------|---|
| BigInsights for Hadoop     | Служба InfoSphere BigInsights, заснована на програмному забезпеченні з відкритим вихідним кодом Hadoop, забезпечує роботу HBase, Hive, MapReduce, Pig і інших програм з відкритим вихідним кодом, в а також призначених для користувача пакетів з відкритим вихідним кодом. |
| Business Rules             | Дозволяє розробникам витратити менше часу на переписування програмного коду і тестування при зміні бізнес-правил. Мінімізує зміни в коді, відокремлюючи бізнес-логіку від логіки застосунку.  |
| Cloudant NoSQL DB          | Забезпечує доступ до розміщеного на серверах постачальника послуг постійно діючого рівня NoSQL JSON-бази даних. Ця служба сумісна з CouchDB і має зручний HTTP-інтерфейс для мобільних і веб-застосунків.   |
| Data Cache                 | Підвищує продуктивність веб-додатків і зручність роботи з ними, витягуючи інформацію з швидких керованих буферів в оперативній пам'яті, які вигідно відрізняються бази даних на жорстких дисках.  |
| Geospatial Analytics       | Реалізує геопросторовий аналіз реального часу для відстеження входу мобільних пристроїв в певні зони і їх виходу з цих зон.   |

|                             |   |
|-----------------------------|---|
| Internet of Things          | Дозволяє застосункам обмінюватися даними з підключеними до Інтернету пристроями, датчиками і шлюзами.   |
| MongoDB                     | Популярна база даних NoSQL  |
| SQL Database                | Додає в застосунок користувача службу реляційної бази даних на вимогу. Ця заснована на DB2 керована служба бази даних працює з веб-і транзакційними застосунками.                                 |
| Watson Machine Translation  | Перекладає текст, введений на тій чи іншій мові, на мову кінцевого користувача. Підтримується переклад між англійською, бразильською португальською, іспанською, французькою та арабською мовами. |
| Watson Personality Insights | Витягує аналітичну інформацію з даних про транзакції і даних соціальних мереж, виявляючи психологічні риси, що визначають рішення про покупки   |

Таблиця 2.1 — Опис сервісів Bluemix

Даний сервісний провайдер є перспективним хмарним середовищем.

### 2.2.2. Microsoft Azure

Microsoft (Windows) Azure — назва хмарної платформи Microsoft. Надає можливість розробки і виконання програм і зберігання даних на серверах, розташованих в розподілених дата-центрах[16]. Microsoft Azure реалізує дві хмарні моделі — платформи як сервісу (PaaS) та інфраструктури як сервісу (IaaS).

Основні особливості даної моделі:

- Оплата тільки за використані ресурси.
- Загальна, багатопотокова структура обчислень.
- Абстракція від інфраструктури.

В основі роботи Microsoft Azure лежить запуск віртуальної машини для кожного екземпляра застосунку. Розробник визначає необхідний обсяг для зберігання даних і необхідні обчислювальні потужності (кількість віртуальних машин), після чого платформа надає відповідні ресурси. Коли початкові потреби в ресурсах змінюються, відповідно до нового запиту замовника платформа виділяє під застосунок додаткові або скорочує невикористовувані ресурси дата-центру.

Microsoft Azure як PaaS забезпечить не тільки всі базові функції операційної системи, але і додаткові: виділення ресурсів на вимогу для необмеженого масштабування, автоматичну синхронну реплікацію даних для підвищення відмовостійкості, обробку відмов інфраструктури для забезпечення постійної доступності та багато іншого.

Microsoft Azure також реалізує інший тип сервісу — інфраструктуру як сервіс. Модель надання інфраструктури (апаратних ресурсів) реалізує можливість оренди таких ресурсів, як сервери, пристрої зберігання даних та мережеве обладнання. Управління всією інфраструктурою здійснюється постачальником, споживач управляє тільки операційною системою і встановленими застосунками.

### **2.2.3. Amazon Web Services**

Amazon Web Services (AWS) — інфраструктура платформ хмарних веб-сервісів, Представлена компанією Amazon на початку 2006 року. У даній інфраструктурі представлено багато сервісів для надання різних послуг, таких як: зберігання даних (файловий хостинг, розподілені сховища даних), Оренда віртуальних серверів, надання обчислювальних потужностей.

Amazon Web Services пропонує широкий набір глобальних сервісів обчислення, сховища, бази даних, аналітики, розгортання, які допомагають організаціям швидше розвиватися, скоротити витрати на ІТ та масштабувати застосунки[17]. Найбільші підприємства і багатообіцяючі стартапи довіряють цим сервісам підтримку роботи широкого спектру функцій, включаючи мобільні та

інтернет застосунки, “Інтернет речей”, розробку ігор, обробку та зберігання даних, сховища, архівацію і багато іншого.

AWS — це модульний конструктор, в якому клієнт сам створює необхідний сервіс з параметрами конфігурації.

Спочатку, платформа AWS замислювалася як доступ до хмарного обчислювального ресурсу. З високою швидкістю доступу і максимальною надійністю зберігання даних, що досягається за рахунок кластеризації. Похідними від цієї ідеї є:

- Віртуальний обчислювальний ресурс.
- Віртуальний виділений сервер.
- Хмарне сховище даних (томи пам'яті для VDS, реляційні бази даних, файлові сховища та ін.).
- Системи аналітики.
- Системи розподіленої доставки контенту (CDN, в термінології системи — Amazon CloudFront).
- Сервіси поштових розсилок.

AWS Elastic Beanstalk — це простий у використанні сервіс для розгортання і масштабування веб-додатків і сервісів, розроблених за допомогою Java, .NET, PHP, Node.js, Python, Ruby, Go і Docker, на серверах Apache, Nginx, Passenger і IIS.

Elastic Beanstalk автоматично виконує розгортання завантаженого коду: виділяє ресурси, займається балансуванням навантаження, автоматичним масштабуванням і моніторингом працездатності програми. При цьому зберігається повний контроль над ресурсами AWS, доступ до них можна отримати в будь-який час.

#### **2.2.4. Force.com**

Force.com- це хмарна платформа для розробки бізнес-рішень від компанії Salesforce.com. Вона дозволяє в мінімальні терміни створювати хмарні бізнес застосунки. Розробка на Force.com — це значно менше зусиль, в порівнянні з

традиційними засобами розробки, так як вимагає менше написання коду і вкрай проста в розгортанні та підтримки. Вся справа в тому, що вона працює за технологією хмарних обчислень (Cloud Computing) і надає безліч зручних інструментів для розробника. Крім того, Force.com бере на себе вирішення питань продуктивності і технічної інфраструктури[18].

На платформі Force.com працює CRM-система Salesforce.com. Всі можливості платформи стають доступні компаніям з покупкою ліцензій даного рішення, починаючи з версії Enterprise Edition (на більш дешевих версіях є ряд обмежень). Також платформа може бути придбана компанією окремо. Це є оптимальним варіантом для проектів, які потребують мінімум стандартної функціональності готового рішення Salesforce, так як ліцензії платформи значно дешевше. Більш того, виробник платформи Salesforce.com дуже гнучко підходить до продажу своїх продуктів, тому можливе придбання декількох типів ліцензій без втрати єдності системи[19]. Наприклад, ліцензії готового рішення Salesforce для підрозділів продажів і сервісної підтримки, а також ліцензії Force.com для департаменту управління персоналом.

### **2.2.5. Порівняльний аналіз хмарних провайдерів**

Були окремо взяті порівняння IBM Bluemix і Microsoft Azure і порівняння IBM Bluemix і Amazon Web Services (таблиця 2.2).

Критеріями порівняння є:

- Рейтинг — загальна оцінка користувачів програмного забезпечення.
- Зручність — зручність і простота користування платформами.
- Підтримка — наскільки розвинена підтримка програмного продукту, як легко можна отримати інформацію і допомогу у вирішенні проблем при розробці.
- Керованість — як багато витрат потрібно докласти для створення, адміністрування і зміни застосунків.
- Модель обслуговування.

- Орієнтовність на розмір компанії — як сильно дане рішення підходить для різних компаній.

Всі оцінки усереднені. Кількість відгуків по кожній з платформ коливається в межах 20-30.

|                           | <i>IBM<br/>Bluemix</i> | <i>Microsoft<br/>Azure</i> | <i>Amazon Web<br/>Services</i> | <i>Force.com</i> |
|---------------------------|------------------------|----------------------------|--------------------------------|------------------|
| Рейтинг                   | 3.7 з 5                | 4.2 з 5                    | 4.6 з 5                        | N / A            |
| Зручність<br>використання | 8.6 з 10               | 8.4 з 10                   | 9.3 з 10                       | N / A            |
| Підтримка                 | 7.1 з 10               | 7.7 з 10                   | 8.7 з 10                       | N / A            |
| Модель<br>обслуговування  | PaaS                   | PaaS, IaaS                 |                                |                  |
| Можливість навчання       | +                      | -                          | +                              | -                |
| Рік запуску               | 2014                   | 2008                       | 2006                           | 2009             |
| Малий бізнес              | 29%                    | 33%                        | 24%                            | N / A            |
| Середній бізнес           | 14%                    | 35%                        | 59%                            | N / A            |
| Підприємства              | 57%                    | 32%                        | 17%                            | N / A            |

Таблиця 2.2 — Порівняльна таблиця IBM Bluemix, Microsoft Azure і Amazon Web Services

Платформа Force.com є свого роду конструктором для створення застосунків в хмарі. Через відсутність гнучкості в процесі розробки, платформа не підходить для навчання.

З таблиці 2.2 видно, що в багатьох параметрах платформи Microsoft Azure і Amazon Web Services виграють у IBM Bluemix.

До явних переваг Microsoft Azure можна віднести реалізацію не тільки PaaS моделі, а й IaaS.

Amazon Web Services має широкий набір глобальних сервісів обчислення, сховищ, баз даних, аналітики для створення і розгортання застосунків.

Також важливою перевагою цих двох платформ є поширеність і перевірка часом (Microsoft Azure існує з 2008 року, а Amazon Web Services з 2006 року).

Таким чином, вибір на користь AWS обумовлений такими факторами:

- Проведення великої кількості майстер-класів для студентів.
- Гібридне хмарне середовище.
- Велика кількість сервісів, що надаються.
- Висока якість надаваних сервісів і можливостей для їх спільного використання.
- Є безкоштовний період.

### **2.3. Мова програмування**

Для реалізації інформаційної системи було вибрано мову програмування Java, з огляду на її кросплатформеність та поширеність. Так як ця мова програмування кросплатформлена, то програми, написані цією мовою, можуть без будь-яких зусиль і оптимізацій запускатися на будь-яких операційних системах. Також через поширеність цієї мови, інформаційну систему можна легко доопрацьовуватися не тільки безпосередньо розробниками системи, а й іншими фахівцями.

Також Java є об'єктно-орієнтованою мовою[20]. ООП дає програмістам три важливі переваги. Перша полягає в спрощенні програмного коду і поліпшення його структуризації. Програми стають простіше для читання і розуміння. Код опису класів, як правило, відділений від коду основної частини програми, завдяки чому над ними можна працювати окремо, все це в свою чергу спрощує процес налагодження та супроводу програми. Друга перевага полягає в тому, що модернізація програм стає незрівнянно більш простим завданням. Найчастіше вона зводиться до додавання нового класу, який успадковує всі властивості одного з наявних класів і містить необхідні додаткові методи[21]. Третя перевага полягає в тому, що одні й ті ж класи можна багато разів використовувати в різних програмах. Вдало створений клас можна зберегти окремо в бібліотечному файлі, і його додавання в програму, як правило, не вимагає внесення серйозних змін в текст.

## 2.4. Апаратна частина СКУД

Система охоронної сигналізації використовується при оснащенні різних типів приміщень. При цьому її метою є припинення будь-якої можливості незаконного проникнення в приміщення, що охороняється або на територію, що охороняється. Основою охоронної системи служать контрольні датчики, які передають інформацію на центральний контрольний пункт. При цьому охоронна сигналізація може бути не тільки автономної, а й функціонувати в комплексі з іншими системами безпеки об'єкту, що охороняється.

Автоматичні системи охоронної сигналізації дозволяють контролювати приміщення, що охороняється або територію 24 години на добу.

До базових компонентів автоматичних систем охоронної сигналізації відносяться:

- Системи пожежної сигналізації — складний комплекс технічних засобів, який реагує на задимлення, зміну температури повітря за допомогою спеціальних датчиків. Датчики посилають сигнал тривоги на приймально-контрольний прилад, після чого сигнал моментально передається на пульт пожежної охорони.
- Системи відеоспостереження, які дозволяють здійснювати відеоконтроль за ситуацією на об'єкті.
- Системи контролю доступу або системи контролю та управління доступом — призначені для відстеження і контролю доступу людей на певну територію. СКД і СКУД можуть бути, як автономними, так і однією зі складових інтегрованої автоматичної системи охоронної сигналізації, що включає перераховані вище компоненти.

В даний час системи управління для більшості комерційних СКУД, а також виконавчі пристрої, що забезпечують рішення задачі управління пристроями, виконуються на основі мікроконтролерів, що мають просту архітектуру, що істотно знижує можливості переналаштування системи, а також прив'язує споживача до конкретного виробника СКУД.

Мною було вибрано 3 можливих варіанти СКУД.

### **2.4.1. Система на базі Arduino**

Істотним фактором є вартість комерційних рішень СКУД, що вимагає чималих матеріальних витрат на розгортання, як самої системи, так і нарощування її можливостей, в разі розширення сфери її застосування.

Тим часом, існує ряд відкритих проєктів по створенню стандартизованих плат програмованих мікроконтролерів, функціональність яких дозволяє розробляти на їх основі виконавчі пристрої СКУД[22].

До таких проєктів, що отримали загальну назву open hardware, відноситься проєкт Arduino — апаратна обчислювальна платформа, основними компонентами якої є мікроконтролер Atmel AVR і середовище розробки на мові Processing / Wiring. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і для підключення до програмного забезпечення, що виконується на комп'ютері[23].

Інформація про схемотехнічну організацію плати знаходиться у відкритому доступі і може бути використана будь-якими розробниками апаратного забезпечення. Завдяки можливості перепрограмування функцій мікроконтролера Arduino, з'являється можливість його застосування в якості основи системи управління СКУД.

Отже, перевагами даної системи, є те, що мікроконтролер Arduino, може бути перепрограмований, відповідно до відкритої документації, яка знаходиться у відкритому доступі.

Недоліком даної системи є низька пропускна можливість.

### **2.4.2. Система на базі StopNet**

Система контролю доступу STOP-Net 4.0 реалізована на платформі інтегрованої системи безпеки (ІСБ) STOP-Net 4.0 і є самостійним продуктом, призначеним для вирішення завдань організації і управління фізичним доступом співробітників і відвідувачів на територію і в окремі приміщення об'єкту. На

поточний момент, в число замовників СКД STOP-Net 4.0 входить понад 4000 організацій різних форм власності, в їх числі — бізнес-центри, державні установи, банки, промислові підприємства, навчальні заклади тощо. Відмінною особливістю системи STOP-Net є баланс між вартістю і функціональними можливостями, висока надійність і стійкість до збоїв, а також продумана архітектура, що дозволяє компаніям-інсталяторам гнучко підходити до проектування рішення для конкретного об'єкта. У нових економічних умовах, коли всі витрати на безпеку повинні бути максимально обгрунтованими, впровадження системи контролю доступу STOP-Net 4.0 дозволяє замовнику в рамках помірною бюджету зменшити втрати, пов'язані з протиправними діями сторонніх осіб або власного персоналу, а постачальникам рішення — реалізувати проект при мінімальних витратах на монтаж і його подальший супровід.

В основі створення ІСБ STOP-Net 4.0 лежать наступні принципи:

- Модульність та гнучкість, що дозволяють проектувати рішення для різних об'єктів з урахуванням їх територіальної організації та особливостей кабельної інфраструктури. При цьому, побудова системи здійснюється на основі широкого переліку спеціалізованих пристроїв, що забезпечують вибір оптимальної конфігурації STOP-Net 4.0 для конкретного замовника за конкурентноздатною ціною.
- Надійність, як стійкий до помилок пакетний протокол обміну даними, а також високий рівень самодіагностики і корекції збоїв в обладнанні та програмних модулях системи. Слід зазначити, що основні операції з обладнанням (налаштування, моніторинг працездатності, завантаження даних) здійснюються у фоновому режимі без збільшення часу реакції системи на зовнішні події.
- Здатність до модернізації, що дозволяє використовувати у складі СКД STOP-Net 4.0 встановлене раніше на об'єкті обладнання СКД STOP-Net 3.5.

Основні можливості ІСБ STOP-Net 4.0:

- Автоматизований контрольований доступ співробітників і відвідувачів на територію та до внутрішніх приміщень об'єкта з урахуванням призначених прав доступу.

- Глобальний контроль повторного проходу, що дозволяє припинити використання перепустки після передачі її стороннім особам або в результаті викрадення.

- Моніторинг об'єкта з оперативним виведенням на екран поста охорони повідомлень про штатні і позаштатні події в системі.

- Управління одноразовими перепустками в рамках повноцінного бюро перепусток з можливістю розміщення заявок, контролювання видачі, обліку реквізитів відвідувачів тощо.

- Моніторинг присутності співробітників і відвідувачів на об'єкті в режимі реального часу.

- Дистанційне розблокування місць доступу під керуванням чергового охоронця в ручному режимі, а також автоматичне керування розблокуванням по команді зовнішніх систем (наприклад, пожежною сигналізацією).

- Наявність власних інструментів для роботи з базами даних (резервування, адміністрування і т. п.).

- Можливість блокування (замикання) окремого приміщення за допомогою пропуску з відповідними правами.

- Повноцінна персоналізація карток доступу, що включає створення і редагування дизайну перепустки, фотозйомку співробітників і безпосередній друк перепустки на термопринтері Evolis.

- Облік робочого часу співробітників і ведення відповідної звітності (звіти щодо порушення трудової дисципліни, таблиці відпрацьованого робочого часу встановленого зразка тощо).

- Підготовка робочої звітності згідно з обраними умовами.

Конфігурацію системи STOP-Net 4.0 для конкретного об'єкта підбирають з урахуванням двох можливих варіантів організації підсистеми обладнання:

- Перетворювач інтерфейсів / універсальний контролер — незалежна конфігурація для об'єктів зі складною політикою безпеки, що вимагають автономності обладнання та зручності його обслуговування.

- Контролер високого рівня / дверний модуль (турнікетний модуль) — оптимальний варіант організації локальних (віддалених) сегментів системи безпеки, що підключаються по мережі Ethernet, а також оснащення об'єктів з великою кількістю однотипних, компактно розташованих місць доступу.

Недолік даної системи полягає в тому, що це є закрита система, з власною інфраструктурою і наразі немає можливості з інтегруватись з нею.

### **2.4.3. Система на базі Milestone та Biostar2**

BioStar 2 Suprema — це відкрита і інтегрована платформа безпеки, заснована на веб-технологіях, яка надає комплексні функціональні можливості для контролю доступу та обліку робочого часу[24].

Дана система володіє модульною, гнучкою структурою. Платформа підтримує як BioStar 2 Device SDK, який використовується для інтеграції терміналів Suprema зі сторонніми системами, так і веб-API, що використовуються для інтеграції платформи BioStar 2 зі сторонніми системами. Крім того, в епоху мобільної ери, існує мобільний застосунок для BioStar 2, який був розроблений, щоб дозволити не тільки дистанційно керувати платформою BioStar 2, але і мати при собі мобільну карту доступу.

Можливості BioStar 2.7.0:

- Інтерактивна графічна карта. Адміністратори можуть відстежувати і контролювати точки доступу за допомогою нового інтерфейсу інтерактивної графічної карти.

- Група необмеженого доступу / рівнів. Для максимальної гнучкості системи BioStar 2 допускає необмежену кількість груп і рівнів доступу.

- Управління часом і відвідуваністю, автоматичне повідомлення. Коли працівник / працівниця переробляє встановлені робочі години, то на електронну пошту адміністратора приходить автоматичне оповіщення.

- Гнучкий робочий графік. З BioStar 2 тепер працівник сам може вибрати робочий розклад.

#### Характеристики:

- Гнучка системна архітектура. Підтримує як централізовані, так і розподілені, створюючи оптимальну систему.
- Оптимальне рішення для контролю доступу. З базовими функціями (управління дверима, ліфтом, контроль різних зон і моніторинг) надаються додаткові функції контролю доступу, такі як графічна карта, узгодження серверів, відео-журнали, журнали зображень і журнал обліку.
- Ведення відеожурналів. Будь-яка подія, яка відбувається у дверей, записується за допомогою мережевого відеореєстратора і IP-камери, які можна відстежити.
- Гнучке рішення для управління часом і відвідуваністю. Можна встановлювати різні правила роботи і створювати персоналізовані звіти про час і відвідуваність, також підтримується перегляд календаря розкладу і налаштування фіксованої / гнучкої зміни.
- Дистанційне керування та мобільні карти. Забезпечує реєстрацію користувачів, оповіщення в режимі онлайн і керування дверима в додатку BioStar 2 Mobile, а також підтримує аутентифікацію за допомогою мобільних карт.
- Підтримує системну інтеграцію. Підтримує інтеграцію платформи на основі RESTful API і апаратну інтеграцію на основі SDK (BioStar 2 API, BioStar 2 Device SDK);
- Підтримує інтеграцію з VMS XProtect Milestone.

Беручи до уваги всі можливості і характеристики системи BioStar2 було прийнято рішення взяти її за основу СКУД в розроблюваній системі “Відкритий спортмайданчик з е-сервісами”.

#### **2.4.4. Зчитувач Suprema Xpass S2**

Suprema XPass S2 — це зчитувач безконтактних карт з функцією контролера в компактному корпусі. Доступні різні модифікації технологій для зчитування карт:

HID Prox, 125kHz EM, Mifare / DesFire. Також модель Xpass S2 підтримує режим використання смартфона в якості ідентифікатора.

Завдяки підтримці протоколу TCP / IP, а також більш традиційних інтерфейсів Wiegand і RS485 досягається гнучкість рішення і можливість роботи з контролерами СКУД різних виробників[25].

Модель Xpass по дизайну близька до біометричних зчитувачів BioEntry Plus і BioEntry W, тим самим відповідає міжнародним стандартам. Ширина корпусу Xpass становить 45 мм: можлива установка практично на будь-яку дверну раму, стійку або нішу.

Зчитувач Xpass S2 з глибиною корпусу менше 12 мм оптимально підходить для установки в стандартному монтажному коробі.

Корпус зчитувачів Xpass / Xpass S2 володіє класом захисту IP65, що дозволяє використовувати пристрій в несприятливих умовах з підвищеними вимогами до пило- та вологозахищеності (рисунок 2.5).



Рисунок 2.5 — Зчитувач Suprema XPass S2.

Кольоровий світлодіодний індикатор і багатотональний зумер забезпечать зрозумілий зворотній зв'язок користувачеві.

Вбудований процесор і великий об'єм пам'яті дозволяє використовувати Xpass / Xpass S2 в автономному режимі “зчитувач / контролер”. Управління та налаштування здійснюються за допомогою спеціалізованого ПЗ Suprema через веб-інтерфейс або за допомогою Suprema SDK.

Беручи до уваги всі характеристики і можливості зчитувача Suprema XPass S2, було прийнято рішення використати його в розроблюваній системі.

### 3. РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

Приклад роботи з системою відображений на діаграмі послідовності (рисунок 3.1).

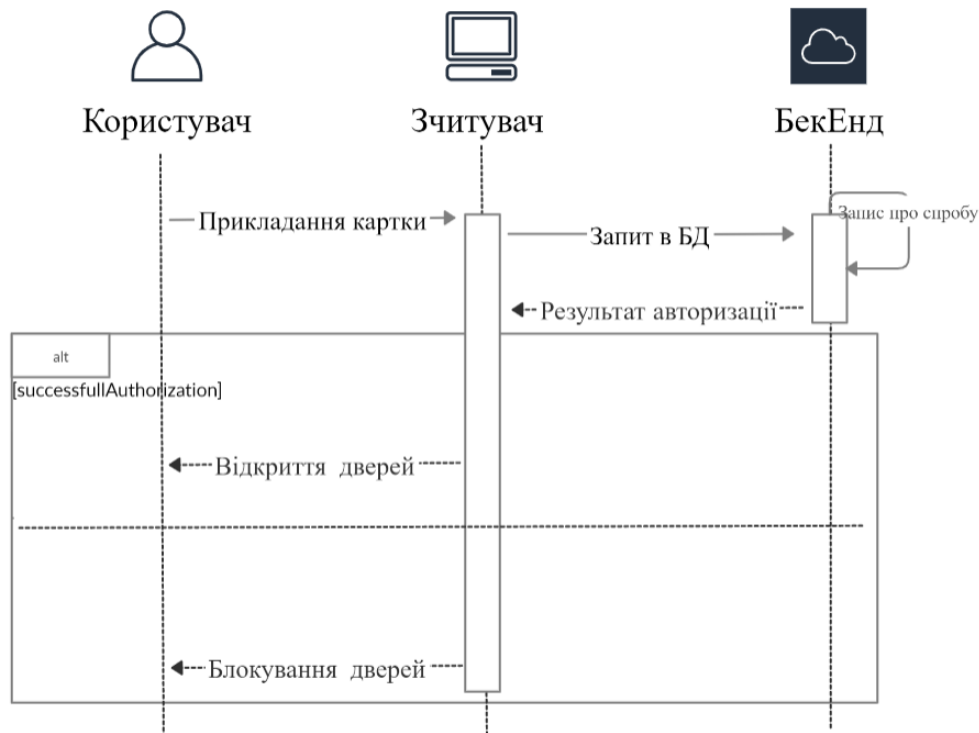


Рисунок 3.1 — Діаграма послідовності роботи системи

Послідовність дій наступна:

- Студент, що бажає потрапити на територію спортивного майданчику, прикладає студентський квиток до зчитувача.
- Зчитувач отримує код карти і передає цю інформацію на контролер.
- Контролер шифрує і передає інформацію про номер дверей, номер картки по бездротовій мережі на сервер.
- Серверне ПЗ отримує інформацію від контролера, виконує дешифрування інформації та робить запит до бази даних, з метою отримати відомості про студента і дозвіл на прохід.

- Звіривши інформацію про студента, серверне ПЗ заносить в базу даних інформацію про те, що студент отримав дозвіл або відмову в проході через двері, а також поточну дату і час. Також інформація передається на контролер.

- Контролер, після отримання відповіді, відправляє на зчитувач сигнал для подачі звукового і світлового сигналу, а також, в разі позитивної відповіді, перемикає реле відмикання замка на певний проміжок часу.

Згідно з діаграми послідовності була розроблена високорівнева архітектура, яка представлена на рисунку 3.2.

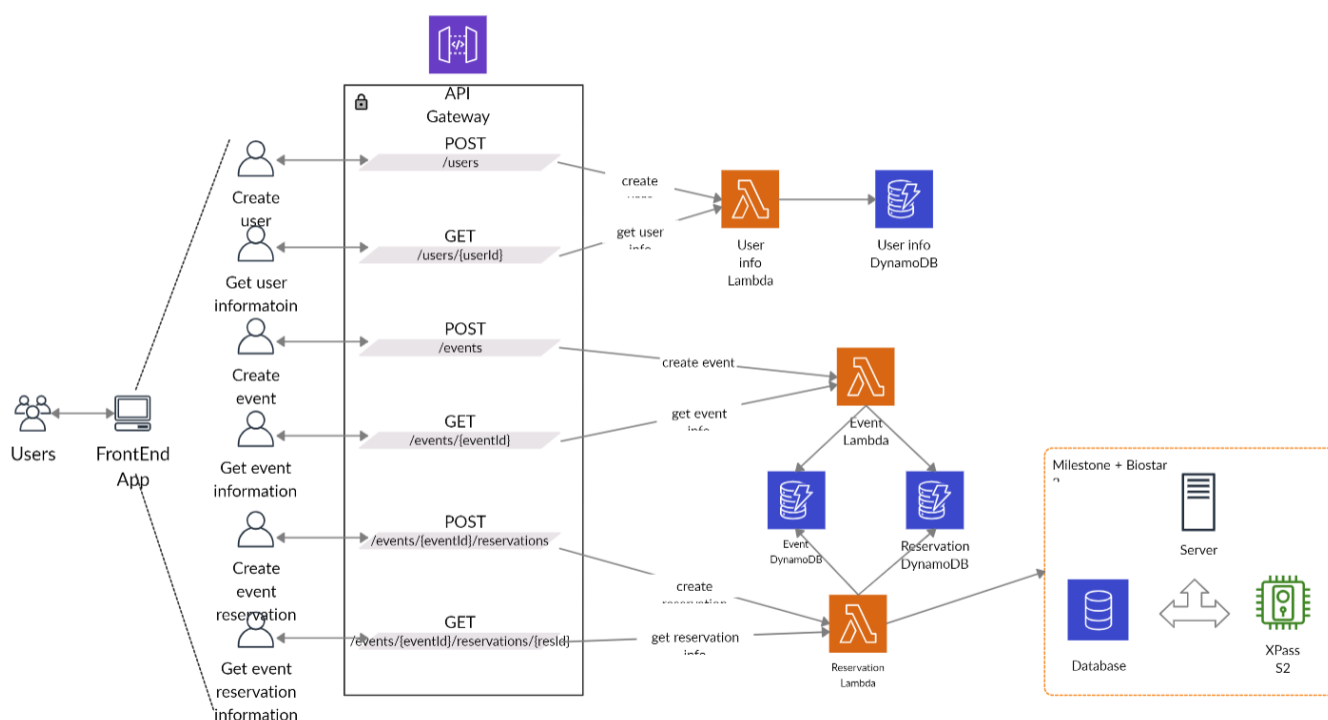


Рисунок 3.2 — Високо рівнева архітектура

Дана діаграма була розроблена з використанням Rest клієнт-серверної архітектури.

### 3.1. Реалізація програмної частини

Виділимо ряд програмних компонентів для аналізу: комплекс програмного забезпечення для програмування контролера, система управління базами даних, обчислювальну платформу. Веб-контейнер AWS Lambda підтримує програмування

на мові Java. Для написання і запису коду в незалежну пам'ять контролера Suprema XPass S2, використовуємо Intelij IDEA та веб-інтерфейс Biostar 2.

З огляду на вимоги роботи системи, розглянемо варіанти СУБД:

- SQLite: з переваг можна виділити зберігання всієї бази даних в одному файлі, можливість швидкого масштабування; до мінусів відносяться відсутність типів даних, що зберігають час і дату, обмежена продуктивність[26].
- MySQL: до плюсів даної системи можна віднести велику кількість підтримуваних форматів даних, велика кількість модулів, багатий функціонал SQL, висока швидкість роботи і безпеку; основним мінусом є ряд функціональних обмежень[27].
- PostgreSQL: система має велику кількість типів даних, ряд унікальних застосунків системи, можливість створювати збережені процедури; великим мінусом є висока навантаження на систему, складністю з пошуком готових серверів з попередньо встановленою системою[28].
- DynamoDB: повністю керована NoSQL база даних, яка підтримує структурну парадигму ключ-значення як для даних, так і для документів[29].

Вибір припав на СУБД DynamoDB завдяки дотриманню всіх вимог до проекту, дешевизні використання.

Таким чином, система що розробляється буде складатися з закритої мережі Wi-Fi, до якої будуть підключені контролер і сервер. На сервері в фоновому режимі буде запущена програма для обробки сигналів з контролера. Для забезпечення безпеки, зв'язок між сервером і контролером відбувається по протоколу HTTPS.

### **3.1.1. Реалізація бази даних**

За допомогою пакету AllFusion ERwin Data Modeler версії 9.7 змодельовано схему DynamoDB бази даних[30]. Схема представлена на рисунку 3.3.

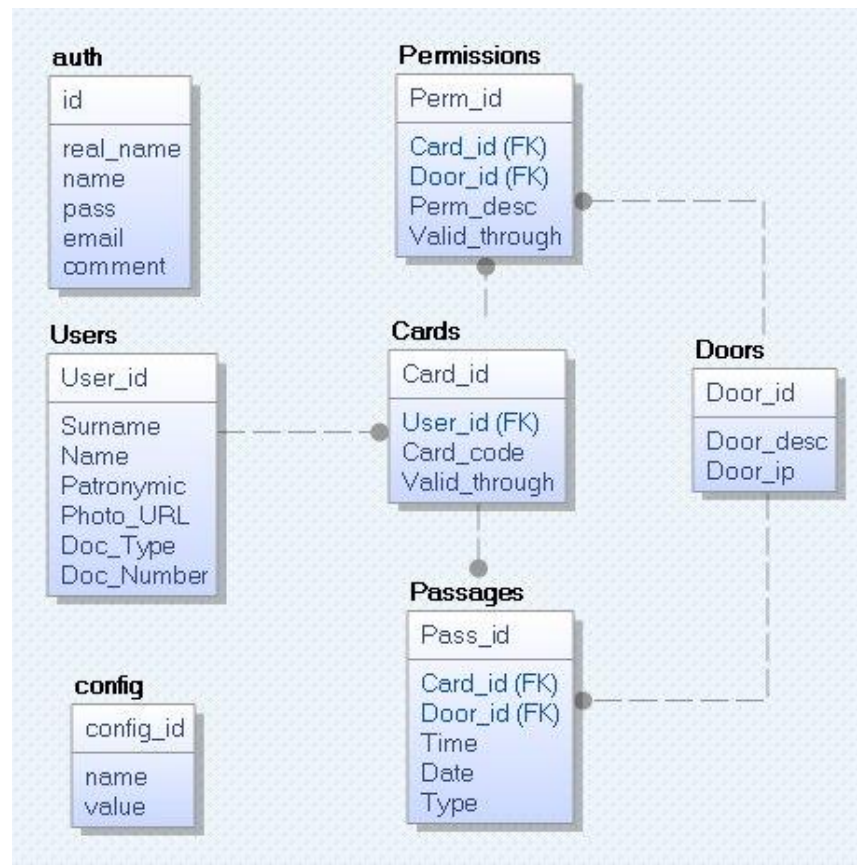


Рисунок 3.3 — Схема бази даних

В даній схемі міститься 7 таблиць, розглянемо докладніше кожен з них:

Таблиці `auth` і `config` містять інформацію для функціонування веб-інтерфейсу. Таблиця `auth` містить відомості про ім'я, псевдонім, електронну адресу та пароль (в зашифрованому вигляді) користувача.

Таблиця `Users` містить інформацію про всіх відвідувачів спортивного майданчику. У ній зберігаються прізвище, ім'я, по батькові і інформація про студентський квиток.

Таблиця `Cards` містить інформацію про студентські квитки. Таблиця зберігає номер карти, номер її власника і термін придатності карти.

Таблиця `Doors` зберігає інформацію про двері, якими управляє СКУД, а точніше номер двері, її опис та ір-адреса контролера, який закріплений за цими дверима.

Таблиця `Permissions` містить інформацію про дозволи на прохід через певні двері студента з певною картою. Проміжок доступу і опис можливо регулювати.

Таблиця Passages містить інформацію про всі проходи студентів через двері. Зберігає номер двері, номер карти, час, день і напрямок руху студента (входить або виходить з майданчику).

Всі відносини автоматично знаходяться в 1 нормальній формі, так як на перетині кожного рядка і кожного стовпця міститься рівно одне значення. Повторюючі рядки відсутні.

Всі відносини мають прості первинні ключі і всі ці відносини перебувають в 1 нормальній формі, значить вони знаходяться і в 2 нормальній формі.

У всіх відношеннях всі не ключові атрибути взаємно незалежні, отже, (з огляду на те, що вони знаходяться в 2 нормальній формі) відносини знаходяться в 3 нормальній формі.

Так як в отриманій моделі немає відносин, у яких був б перетин потенційних ключів, відносини задовольняють нормальній формі Бойса-Кодда.

Розроблена модель не містить сутностей, що мають три ключових атрибута. Таким чином, всі сутності автоматом знаходяться в 4 і 5 нормальних формах.

Для збільшення швидкодії системи, створено унікальний індекс в таблиці Users на стовпець Doc\_Number, в таблиці auth на стовпці email і name, в таблиці Card на стовпець Card\_code і в таблиці config на стовпець name.

Основними запитами до бази є запити по полю Card\_code дізнатися Card\_id і по полю Card\_id дізнатися Perm\_id. Для прискорення роботи цих запитів, створено індекси в таблиці Cards на стовпець Card\_code і в таблиці Permissions на стовпець Perm\_id.

### **3.1.2. Реалізація серверної програми**

Сервіси, що розгорнуті на обчислювальній платформі AWS Lambda, підключені до Api Gateway. У разі отримання нового повідомлення сервіс виконує запит до бази даних DynamoDB для отримання номера карти по її коду, перевіряє дозвіл на вхід в і вихід з майданику, шифрує вихідний рядок з додаванням даних про доступ і відправляє назад на контролер. Також присутні CRUD сервіси для всіх



## 3.2. Реалізація апаратної частини

### 3.2.1. Підключення зчитувача до системи

Схема підключення контроллера показана на рисунку 3.5.

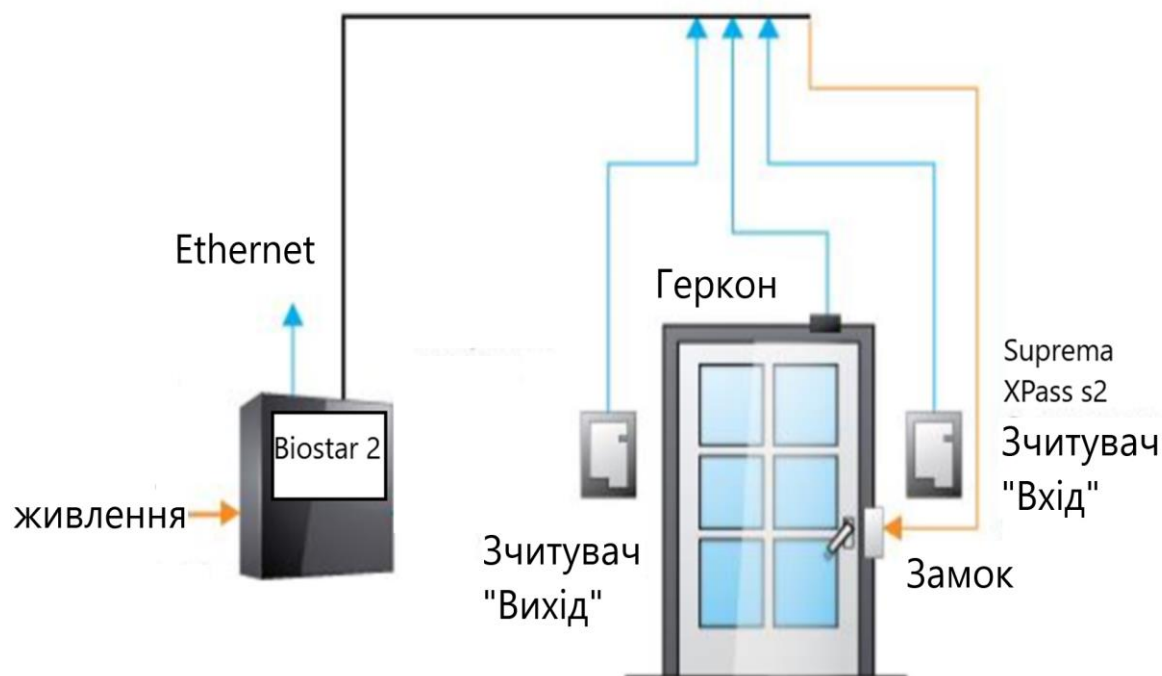


Рисунок 3.5 — Схема підключення зчитувача Suprema XPass S2

Послідовність дій наступна:

1) Монтаж кронштейна і зчитувача.

- За допомогою кріпильних гвинтів був встановлений кронштейн.
- Зчитувач Xpass S2 був встановлений на змонтований кронштейн.
- Зчитувач Xpass S2 був під'єднаний до кронштейну за допомогою загвинчування кріпильних гвинтів на нижній частині Xpass S2.

2) Підключення до живлення.

- Було використано окреме джерело живлення для безпечного введення / виведення електричного замка і Xpass S2 відповідно.

3) Підключення до мережі.

- Зчитувач був підключений до локальної мережі через підключення до маршрутизатора.

### 3.2.2. Інтеграція BioStar та VMS Milestone

Процедура інтеграції BioStar та VMS Milestone наступна (рисунок 3.6):

- Встановлено BioStar 2 на VDS (віртуальний сервер).
- Встановлено Milestone XProtect VMS з професійною ліцензією.
- Встановлені плагіни для інтеграції BioStar 2 для Milestone XProtect.
- Додані зчитувачі в Milestone XProtect VMS.
- Створено нову консоль доступу для підключення BioStar 2.

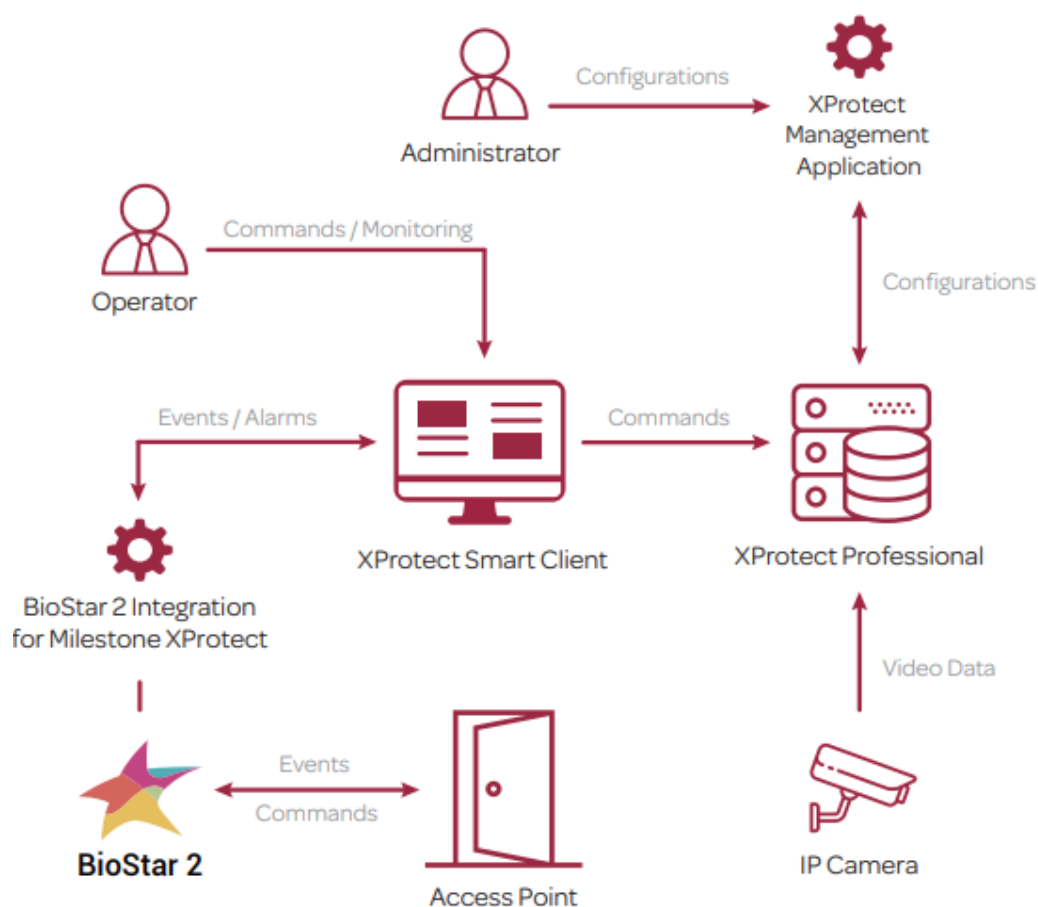


Рисунок 3.6 — Інтеграція BioStar 2 та VMS Milestone

Після цих дій смарт-клієнт Xprotect має змогу керувати програмним забезпеченням BioStar 2.

## 4. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З СИСТЕМОЮ

Для забезпечення безвідмовної роботи програмної системи контролю доступу треба дотримуватися основних вимог та рекомендацій щодо її використання.

Для користування розробленою програмною системою персональний комп'ютер повинен мати доступ в мережу. Для використання апаратно частини користувач повинен мати картку з RFID міткою (студентський квиток).

### 4.1. Використання програмної частини сервісу

Для використання програмної частини сервісу необхідно встановити та відкрити середовище розробки API (рисунок 4.1), наприклад Postman.

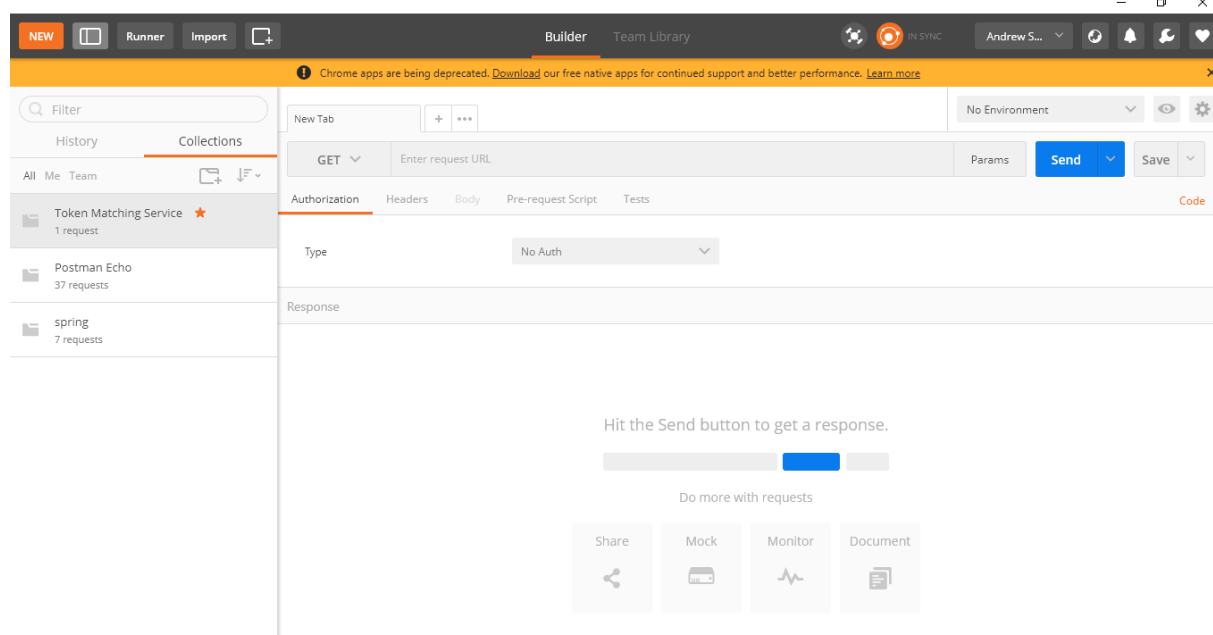


Рисунок 4.1 — Середовище розробки API Postman

Postman — це повний набір інструментів для розробників API, який використовується більш ніж 3 мільйонами розробників та 30 тисячами компаній по всьому світу[31].

Postman робить роботу з API швидшою і простішою, підтримуючи

розробників на кожному етапі їх робочого процесу, і доступний для користувачів Mac OS X, Windows, Linux та Chrome.

Також, можна скористатися інструментом Swagger Editor.

Далі потрібно створити POST запит, вказавши адрес сервісу та тіло запиту (рисунок 4.2).

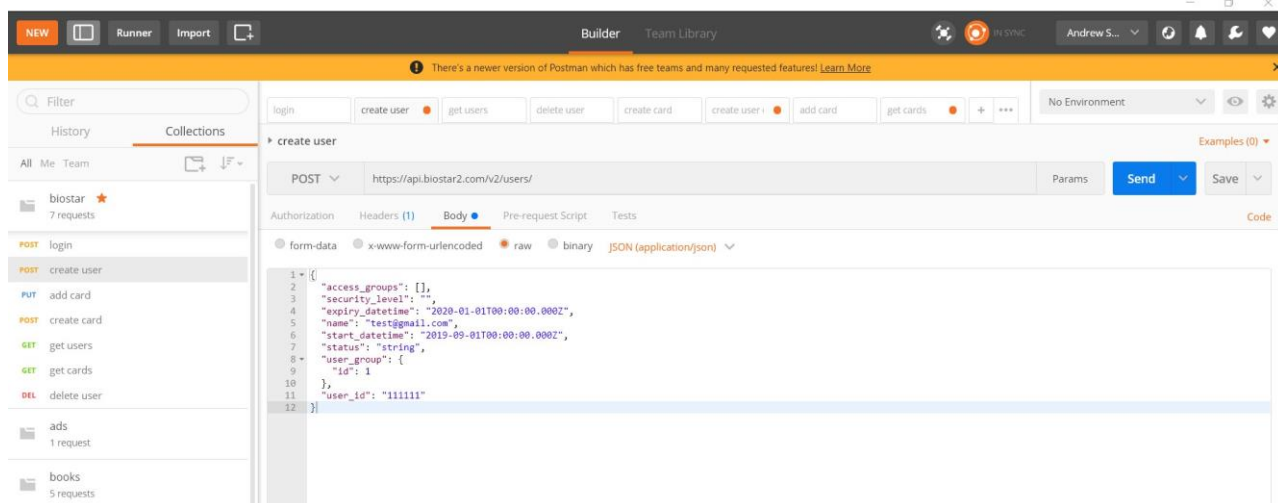


Рисунок 4.2 — POST запит створення користувача СКУД

Далі потрібно натиснути клавішу “Send”. У вікні відповіді з’явиться повідомлення від серверу у форматі JSON з HTTP статусом кодом 200. (рисунок 4.3)

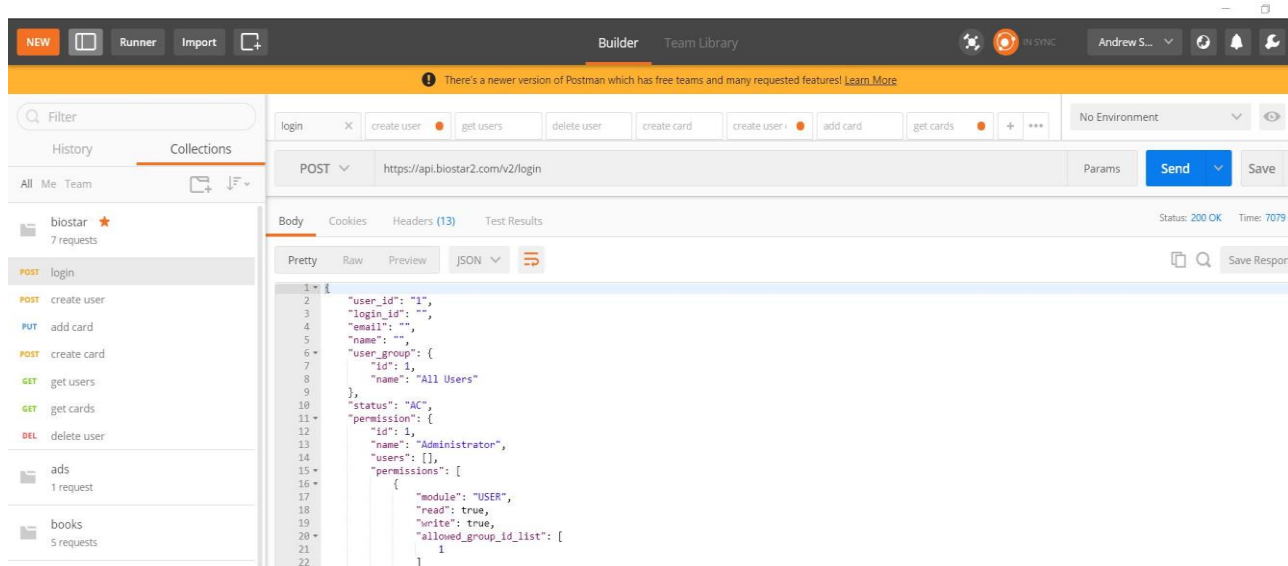


Рисунок 4.3 — Відповідь сервера у форматі JSON

Використовуючи середовища розробки API, можна конструювати REST запити до програмно-апаратної частини сервісу: створювати, видаляти, змінювати користувачів, спортивних заїходів, тощо. Спостерігати за метаданими запиту, а

саме — швидкість і розмір відповіді серверу.

## 4.2. Використання апаратної частини сервісу

Для використання контролера потрібно піднести картку з рфід міткою (студентський квиток) до рфід сканера (рисунок 4.4).



Рисунок 4.4 — Робота контролера Suprema XPass S2

Зліва на русинку показано відкриття дверей, яке супроводжується звуковим сигналом і зеленим кольором світлодіоду, справа – блокування, яке супроводжується звуковим сигналом і червоним кольором світлодіоду.

## 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

В даному розділі описано економічне обґрунтування і доцільність створення стартап проекту на основі програмно-апаратного комплексу.

### 5.1. Опис ідеї стартап-проекту

Розділ має на меті проведення маркетингового аналізу стартап проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження.

Метою розділу є формування інноваційного мислення, підприємницького духу та формування здатностей щодо оцінювання ринкових перспектив і можливостей комерціалізації основних науково-технічних розробок, сформованих у попередній частині магістерської дисертації у вигляді розроблення концепції стартап-проекту в умовах висококонкурентної ринкової економіки глобалізаційних процесів. Опис стартап-проекту наведено у таблиці 5.1.

| <i>Зміст ідеї</i>   | <i>Напрямки застосування</i>   | <i>Вигоди для користувача</i>  |
|---|--|--|
| Створення та розгортання бази знань у хмарі з наданням користувачеві доступу читання та запису через REST арі. Створення інтеграції системи Biostar 2 та VMS Milestone. | 1. Використання для виконання пропуску студентів по студентському квитку.                          | REST API надає можливість використовувати систему як сервіс, що значно спрощує доступ до функціоналу.  |
|   | 2. Використання інтеграції Biostar та VMS Milestone у інших додатках та системах контролю доступу. | Можливість використання інтеграції Biostar та VMS Milestone дає можливість розробки плагінів та їх оновлення. Сервіс легко інтегрувати у інші системи. |

Таблиця 5.1 — Опис ідеї стартап-проекту

Отже, проект може бути використаний як інструмент для пропуску студентів. Також може бути використаний в інших системах контролю доступу як сервіс з зручним REST API.

| №<br>п/<br>п | Техніко-<br>економічні<br>характери-<br>стики ідеї | (потенційні)<br>товари/концепції<br>конкурентів |             |             |             | W<br>(слабка<br>сторона) | N<br>(нейтральна<br>сторона) | S<br>(сильна<br>сторона) |
|--------------|--|---|-------------|-------------|-------------|--------------------------|------------------------------|--------------------------|
|              |  | Мій<br>проект                                   | Конкурент 1 | Конкурент 2 | Конкурент 3 |                          |                              |                          |
| 1.           | Форма виконання                                    | Веб-сервіс                                      | Програма    | Веб-додаток | Програма    |                          |                              | +                        |
| 2.           | Собівартість                                       | Низька  | Низька      | Висока      | Висока      |                          |                              | +                        |
| 3.           | Кросплатформність                                  | Так   | Ні          | Так         | Так         |                          |                              | +                        |
| 4.           | Наявність Rest API                                 | Так   | Так         | Так         | Так         |                          |                              | +                        |
| 5.           | Захищеність Rest API                               | Так   | Ні          | Ні          | Так         |                          | +                            |                          |
| 6.           | Горизонтальне масштабування                        | Так   | Так         | Ні          | Так         |                          | +                            |                          |
| 7.           | Аварійне відновлення                               | Ні  | Ні          | Так         | Ні          | +                        |                              |                          |

Таблиця 5.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Сильними сторонами проекту є низька собівартість, кросплатформність, горизонтальне масштабування, наявність Rest API. Нейтральна сторона — захищеність Rest API. Слабкою стороною проекту є відсутність аварійного відновлення. Отже, система є конкурентноспроможною.

## 5.2. Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту та технології створення товару. (таблиця 5.3).

| <i>№ n/n</i> | <i>Ідея проекту</i>                                   | <i>Технології<br/>її<br/>реалізації</i> | <i>Наявність<br/>технолог<br/>її</i> | <i>Доступність<br/>технологій</i> |
|--------------|---|---|--------------------------------------|-----------------------------------|
| 1.           | Створення<br>СКУД                                     | Arduino                                 | Наявна                               | Безкоштовна,<br>доступна          |
|              |   | Biostar 2                               | Наявна                               | Частково<br>безкоштовна           |
| 2.           | Створення REST<br>API                                 | Spring<br>Boot,<br>Maven                | Наявні                               | Безкоштовна,<br>доступна          |
|              |   | Amazon API<br>Gateway,<br>AWS<br>Lambda | Наявна                               | Платні                            |
| 3.           | Розгортання<br>застосунку в<br>хмарному<br>середовищі | Heroku                                  | Наявна                               | Безкоштовна,<br>доступна          |
|              |   | AWS CDK                                 | Наявна                               | Безкоштовна                       |

Таблиця 5.3 — Технологічна здійсненність ідеї проекту

Обрані технології реалізації ідеї проекту: Biostar 2 через великий функціонал фреймворку та наявність докладної документації, наявність досвіду роботи розробників з даною технологією; Amazon API Gateway, AWS Lambda через простоту використання, можливість розгортання додатків на основі таких технологій у хмарі; AWS CDK для розгортання у хмарі через безкоштовність.

### 5.3. Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів (таблиця 5.4).

| № п/п | Показники стану ринку (найменування)                     | Характеристика                                |
|-------|--|---|
| 1.    | Кількість головних гравців, од                           | 5   |
| 2.    | Загальний обсяг продаж, грн/ум.од                        | 8000 грн./ум.од                               |
| 3.    | Динаміка ринку (якісна оцінка)                           | Зростає                                       |
| 4.    | Наявність обмежень для входу (вказати характер обмежень) | Немає   |
| 5.    | Специфічні вимоги до стандартизації та сертифікації      | Немає   |
| 6.    | Середня норма рентабельності в галузі (або по ринку), %  | $R = (3000000 * 100) / (1000000 * 12) = 25\%$ |

Таблиця 5.4 — Попередня характеристика потенційного ринку стартап-проекту

Отже, було проаналізовано наявність попиту, обсяг, динаміку розвитку ринку.

Обмеження для входу на ринок відсутні, динаміка ринку зростає, галузь є рентабельною.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (табл. 5.5).

| №<br>п/<br>п | Потреба, що<br>формує ринок   | Цільова<br>аудиторія<br>(цільові сегменти<br>ринку)   | Відмінності у<br>поведінці різних<br>потенційних<br>цільових груп<br>клієнтів | Вимоги<br>споживачів<br>до товару   |
|--------------|---|---|---|---|
| 1.           | Необхідне програмне забезпечення (REST API) для систем контролю доступу | Потенційними цільовими групами є університети та компанії, яким потрібна система контролю доступу, яка має можливість масштабування | Цільова група має великий обсяг користувачів системи контролю доступу         | Рішення повинне бути придатним до інтеграції в інші більш складні системи, мати Rest API, бути здатним масштабуватися, бути розгорнутим у хмарі |

Таблиця 5.5 — Характеристика потенційних клієнтів стартап-проекту

Згідно проведеної характеристики потенційних клієнтів стартап-проекту впливає, що на ринку є затребуваним програмне забезпечення (REST API) систем контролю доступу як до сервісу і потенційними цільовими групами є університети та компанії, яким потрібна система контролю доступу, яка має можливість масштабування.

Дане рішення повинне бути придатним до інтеграції в інші більш складні системи.

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 5.6-5.7).

Фактори та загрози в таблиці подаються в порядку зменшення значущості.

| <i>№<br/>п/<br/>п</i> | <i>Фактор</i>   | <i>Зміст загрози</i>   | <i>Можлива реакція компанії</i>  |
|-----------------------|---|--|--|
| 1.                    | Конкуренція   | Вихід на ринок великої компанії                                    | 1. Вихід з ринку<br>2. Запропонувати великій компанії поглинути себе<br>3. Передбачити додаткові переваги власного ПЗ для того, щоб повідомити про них саме після виходу міжнародної компанії на ринок |
| 2.                    | Зміна потреб користувачів                               | Користувачам необхідне програмне забезпечення з іншим функціоналом | 1. Передбачити можливість додавання нового функціоналу до створеного ПЗ  |
| 3.                    | Зміна тарифів провайдера хмарного розгортання на платні | Необхідність оплати послуг провайдера хмари                        | 1. Пошук іншого безкоштовного провайдера<br>2. Пошук інвестицій для оплати існуючого провайдера  |
| 4.                    | Надходження на ринок альтернативних продуктів           | Перехід користувачів нашого товару на інший продукт                | Впровадження нового функціоналу, якого немає у конкурентів   |
| 5.                    | Уповільнення росту ринку                                | Скорочення користувачів продуктів, що тільки виходять на ринок     | Інвестиції у впровадження ефективної реклами продукту  |

Таблиця 5.6 — Фактори загроз

Отже, було проаналізовано фактори загроз ринкового впровадження проекту, серед яких: конкуренція, уповільнення росту ринку, зміна потреб користувачів,

зміна тарифів провайдера хмарного розгортання на платні та надходження на ринок альтернативних продуктів. Було також запропоновано можливі реакції компанії.

| <i>№<br/>n/n</i> | <i>Фактор</i>                                    | <i>Зміст можливості</i>   | <i>Можлива реакція компанії</i>   |
|------------------|--|---|---|
| 1.               | Стрімкий ріст попиту на системи контролю доступу | Наявність попиту на системи контролю доступу                            | Змога запропонувати продукт більшої кількості потенційних користувачів  |
| 2.               | Поява нових ризонерів                            | Надання нового функціоналу для контролю над системою контролю доступом  | Розробка нового функціоналу у вигляді нового HTTP запиту для надання користувачам контролю над системою контролю доступом |
| 3.               | Стрімке зростання росту ринку                    | Компаніям, що тільки виходять на ринок, буде простіше отримати клієнтів | Змога запропонувати продукт більшої кількості потенційних користувачів  |
| 4.               | Обслуговування додаткових груп споживачів        | Поява нових потенційних груп споживачів                                 | Змога розширити продукт для подальшого впровадження у нові галузі   |
| 5.               | Розширення асортименту можливих послуг           | Поява нового функціоналу, що привабить нових користувачів               | Розробка нового функціоналу, що є потребою певної групи користувачів  |

Таблиця 5.7 — Фактори можливостей

У Таблиці 5.7 наведено фактори можливостей ринкового впровадження проекту, серед яких: стрімкий ріст попиту на системи контролю доступу, поява нових ризонерів, стрімке зростання росту ринку, обслуговування додаткових груп споживачів, розширення асортименту можливих послуг; було також запропоновано можливі реакції компанії.

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (таблиця 5.8).

| <i>Особливості конкурентного середовища</i>        | <i>В чому проявляється дана характеристика</i>                        | <i>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</i>   |
|--|---|---|
| 1. Вказати тип конкуренції — досконала             | Існує 3 фірми-конкурентки на ринку                                    | Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами) |
| 2. За рівнем конкурентної боротьби — міжнародний   | Одна з компаній – з іншої країни, дві – з України                     | Додати можливість вибору мови ПЗ, щоб легше було у майбутньому вийти на міжнародний ринок   |
| 3. За галузевою ознакою — внутрішньогалузева       | Конкуренти мають ПЗ, яке використовується лише всередині даної галузі | Створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях                                |
| 4. Конкуренція за видами товарів: — товарно-видова | Види товарів є однаковими, а саме — програмне забезпечення            | Створити ПЗ, враховуючи недоліки конкурентів  |
| 5. За характером конкурентних переваг — нецінова   | Вдосконалення технології створення ПЗ, щоб собівартість була нижчою   | Використання менш дорогих технологій для розробки, ніж використовують конкуренти  |
| 6. За інтенсивністю — не марочна                   | Бренди відсутні   | -   |

Таблиця 5.8 — Ступеневий аналіз конкуренції на ринку

У Таблиці 5.8 наведено ступеневий аналіз конкуренції на ринку, де було визначено особливості конкурентного середовища та їх вплив а діяльність підприємства. Однією з найбільш важливих дій компанії для досягнення конкурентоспроможності є необхідність створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (таблиця 5.9).

| <i>Складові аналізу</i> | <i>Прямі конкуренти в галузі</i>   | <i>Потенційні конкуренти</i>   | <i>Постачальники</i>                         | <i>Клієнти</i>                                 | <i>Товари-замінники</i>  |
|-------------------------|--|--|--|--|--|
|                         | <i>Навести перелік прямих конкурентів</i>  | <i>Визначити бар'єри входження в ринок</i>   | <i>Визначити фактори сили постачальників</i> | <i>Визначити фактори сили споживачів</i>       | <i>Фактори загроз з боку замінників</i>  |
| <b>Висновки:</b>        | Існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 2, так як його рішення також представлене у вигляді веб-додатку | Так, можливість для входу на ринок є, бо наше рішення має Rest API Для доступу та контролю системи контролю доступом | Постачальники відсутні.                      | Важливим для користувача є швидкість роботи ПЗ | Товари-замінники можуть використати більш дешеву технологію створення ПЗ та зменшити собівартість товару |

Таблиця 5.9 — Аналіз конкуренції в галузі за М. Портером

Було здійснено аналіз конкуренції в галузі за М. Портером, в результаті чого було визначено, що існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 2, так як його рішення також представлене у вигляді веб-застосунку, але

можливості для входу на ринок є, бо наше рішення має має Rest API Для доступу та контролю системи контролю доступом.

За результатами аналізу таблиці робиться висновок щодо принципової можливості роботи на ринку з огляду на конкурентну ситуацію. Також робиться висновок щодо характеристик (сильних сторін), які повинен мати проект, щоб бути конкурентоспроможним на ринку.

Другий висновок враховується при формулюванні переліку факторів конкурентоспроможності. На основі аналізу конкуренції, а також із урахуванням характеристик ідеї проекту, вимог споживачів до товару та факторів маркетингового середовища визначається та обґрунтовується перелік факторів конкурентоспроможності (таблиця 5.10).

| <i>№ n/n</i> | <i>Фактор конкурентоспроможності</i>            | <i>Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)</i> |
|--------------|---|--|
| 1.           | Можливість керування системою контролю доступом | Дозволяє користувачам здійснювати контроль через веб-інтерфейс Biostar 2                                   |
| 2.           | Наявність REST API                              | Дозволяє інтегрувати сервіс у складні системи завдяки універсальному API                                   |
| 3.           | Хмарне розгортання                              | Дозволяє звертатись до додатку як до сервісу   |
| 4.           | Горизонтальне масштабування                     | Можливість гнучкого масштабування за допомогою додавання апаратних компонентів                             |
| 5.           | Аварійне відновлення                            | Можливість аварійного відновлення застосунку   |

Таблиця 5.10 — Обґрунтування факторів конкурентоспроможності

У таблиці 5.10 наведено обґрунтування факторів конкурентоспроможності, серед яких: можливість керування системою контролю доступом, наявність REST API та хмарне розгортання. Було також наведено обґрунтування цих факторів.

За визначеними факторами конкурентоспроможності (таблиця 5.10) проводиться аналіз сильних та слабких сторін стартап-проекту (таблиця 5.11).

У наступній таблиці наведено проведення аналізу сильних та слабких сторін стартап-проекту, факторами конкурентоспроможності виступили такі: можливість керування системою контролю доступом, наявність REST API, хмарне розгортання, горизонтальне масштабування, аварійне відновлення.

| № п/п | Фактор конкурентоспроможності                   | Бал и 1-20 | Рейтинг товарів-конкурентів у порівнянні з нашим підприємством |    |    |   |   |   |   |
|-------|---|------------|--|----|----|---|---|---|---|
|       |   |            | -3   | -2 | -1 | 0 | 1 | 2 | 3 |
| 1.    | Можливість керування системою контролю доступом | 20         |  |    | +  |   |   |   |   |
| 2.    | Наявність REST API                              | 15         | +  |    |    |   |   |   |   |
| 3.    | Хмарне розгортання                              | 15         |  | +  |    |   |   |   |   |
| 4.    | Горизонтальне масштабування                     | 10         |  |    |    |   | + |   |   |
| 5.    | Аварійне відновлення                            | 10         |  |    |    |   | + |   |   |

Таблиця 5.11 — Порівняльний аналіз сильних та слабких сторін проекту

Отже, серед сильних сторін проекту можна виділити наступні: можливість керування системою контролю доступом, наявність REST API, можливість хмарного розгортання. Серед слабких сторін можна виділити відсутність можливості аварійного відновлення.

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

У наступній таблиці буде проілюстровано SWOT-аналіз стартап-проекту, тобто його слабкі та сильні сторони, можливості та загрози виходу на ринок.

|  |   |
|--|---|
| Сильні сторони: наявність можливість керування системою контролю доступом, наявність REST API, можливість хмарного розгортання   | Слабкі сторони: можливість зміни тарифів провайдером хмарного розгортання на платні   |
| Можливості: стрімкий ріст попиту на системи контролю доступом, можливість впровадження нових різонерів, стрімке зростання росту ринку, обслуговування додаткових груп споживачів, розширення асортименту можливих послуг | Загрози: конкуренція, зміна потреб користувачів, зміна тарифів провайдера хмарного розгортання на платні, надходження на ринок альтернативних продуктів, уповільнення росту ринку |

Таблиця 5.12 — SWOT-аналіз стартап-проекту

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів.

| № п/п | Альтернатива (орієнтовний комплекс заходів) ринкової поведінки      | Ймовірність отримання ресурсів | Строки реалізації |
|-------|---|--------------------------------|-------------------|
| 1.    | Створення додатку з використанням Biostar 2, AWS Lambda             | 90%                            | 3 місяці          |
| 2.    | Створення програми на основі без використання будь-яких фреймворків | 35%                            | 8 місяців         |

Таблиця 5.13 — Альтернативи ринкового впровадження стартап-проекту

З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими. Тому обираємо альтернативу (створення додатку з використанням Biostar 2, AWS Lambda).

#### 5.4. Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 5.14).

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

| <i>№ n/n</i>   | <i>Опис профілю цільової групи потенційних клієнтів</i> | <i>Готовність споживачів сприйняти продукт</i> | <i>Орієнтовний попит в межах цільової групи (сегменту)</i> | <i>Інтенсивність конкуренції в сегменті</i>                               | <i>Простота входу у сегмент</i>  |
|--|---|--|--|---|--|
| 1.   | Університети  | Спрощення роботи з системами контролю доступу  | Великий  | Існує 3 конкуренти, які надають схожі, але більш вузькі і дорогі рішення. | Наявність REST API   |
| 2.   | Підприємства  | Спрощення роботи з системами контролю доступу  | Великий  |   | Можливість інтеграції в уже існуючі системи завдяки REST API, зручне хмарне розгортання. |
| Які цільові групи обрано: обираємо підприємства та дослідницькі центри |   |  |  |   |  |

Таблиця 5.14 — Вибір цільових груп потенційних споживачів

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку. За М. Портером, існують три базові стратегії розвитку, що відрізняються за ступенем охоплення цільового ринку та типом конкурентної переваги, що має бути реалізована на ринку (за витратами або визначними якостями товару).

Отже, проілюструвати базову стратегію розвитку можна у вигляді таблиці 5.15.

| <i>№<br/>п/<br/>п</i> | <i>Обрана<br/>альтернатива<br/>розвитку<br/>проекту</i>     | <i>Стратегія<br/>охоплення<br/>ринку</i> | <i>Ключові<br/>конкурентоспро-<br/>мо жні позиції<br/>відповідно до<br/>обраної<br/>альтернативи</i> | <i>Базова<br/>strate-<br/>гія<br/>розвитк<br/>у</i> |
|-----------------------|---|--|--|---|
| 1.                    | Створення веб-сервісу, використовуючи Biostar 2, AWS Lambda | Ринкове позиціонування                   | Можливість інтеграції в уже існуючі системи завдяки REST API, зручне хмарне розгортання              | Диференціація                                       |

Таблиця 5.15 — Визначення базової стратегії розвитку

Було обрано таку альтернативу розвитку проекту: створення веб-сервісу, використовуючи Biostar 2, AWS Lambda, адже завдяки цим технологіям можна досягнути ключових конкурентноспроможних позицій кінцевого продукту, а саме: можливість інтеграції в уже існуючі системи завдяки Rest API, використання хмарних технологій.

Далі про визначення базової стратегії конкурентної поведінки (таблиця 5.16). Будуть розглянуті питання першопрохідця на ринку, пошуку нових споживачів або відбору існуючих у конкурентів, копіювання основних характеристик товару конкурента, а також стратегії конкурентної поведінки.

| <i>№ п/п</i> | <i>Чи є проект першопрхідцем на ринку?</i> | <i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</i> | <i>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</i>   | <i>Стратегія конкурентної поведінки</i> |
|--------------|--|---|--|---|
| 1.           | Ні   | Так   | Буде, а саме: основною задачею є розробка ПЗ і інтеграція з системою контролю доступом(конкуренти 1, 2, 3), форма виконання - веб-сервіс (конкурент 2) | Зайняття конкурентної ніші              |

Таблиця 5.16 — Визначення базової стратегії конкурентної поведінки

Отже, було визначено базову стратегію конкурентної поведінки як зайняття конкурентної ніші.

Визначимо стратегію позиціонування у таблиці 5.17, що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

| <i>№ п/п</i> | <i>Вимоги до товару цільової аудиторії</i>               | <i>Базова стратегія розвитку</i> | <i>Ключові конкурентоспроможні і позиції власного стартап-проекту</i>                   | <i>Вибір асоціацій, які мають сформувану позицію власного проекту (три ключових)</i> |
|--------------|--|----------------------------------|---|--|
| 1.           | Наявність універсального API, зручне хмарне розгортання. | Диференціація                    | Можливість інтеграції в уже існуючі системи завдяки REST API, зручне хмарне розгортання | Інтеграція, хмарне розгортання   |

Таблиця 5.17 — Визначення стратегії позиціонування

Отже, було вибрано такі асоціації, які мають сформувати комплексну позицію власного проекту: інтеграція (адже завдяки REST API сервіс просто інтегрувати у існуючі системи), хмарне розгортання (оскільки AWS Lambda легко розгортаються в хмарах)

## 5.5. Розробка маркетингової програми

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 5.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

| <i>№ п/п</i> | <i>Потреба</i>                                 | <i>Вигода, яку пропонує товар</i>  | <i>Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)</i> |
|--------------|--|--|---|
| 1.           | Наявність універсального API                   | Додаток реалізований у вигляді RESTful сервісу, що надає відповіді у вигляді JSON, що дає змогу користувачам звертатись до сервісу за допомогою стандартних HTTP POST та GET запитів | Перевага в універсальності на можливості інтегрувати сервіс у існуючі системи.      |
| 2.           | Можливість зручного хмарного розгортання       | Можливість розгорнути додаток всюди, де є функціонал розгортання AWS Lambda  | Користувачі мають змогу працювати з системою віддалено у хмарі                      |
| 3.           | Можливість керування системою контролю доступу | Можливість керування системою контролю доступу через веб-інтерфейс   | Веб-інтерфейс   |

Таблиця 5.18 — Визначення ключових переваг концепції потенційного товару

Отже бачимо, що проект має ключові переваги перед конкурентами, які повністю відповідають потребам цільової аудиторії.

Застосунок реалізований у вигляді RESTful сервісу, що надає відповіді у вигляді JSON, що дає змогу користувачам звертатись до сервісу за допомогою стандартних HTTP POST та GET запитів, а це є досить універсальним способом для подальшої інтеграції сервісу в інші системи.

Далі у таблиці 5.19 проілюстрована трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання.

| <i>Рівні товару</i>  | <i>Сутність та складові</i>   |      |                |
|--|---|------|----------------|
| I. Товар за задумом  | Веб-сервіс, що надає доступ до системи контролю доступом за допомогою HTTP запитів, надає можливість пропуску користувачів по рфід картам |      |                |
| II. Товар у реальному виконанні  | Властивості/характеристики  | М/Нм | Вр/Тх /Тл/Е/Ор |
|  | 1. Наявність універсального API   | 1.Нм | 1.Технологічна |
|  | 2. Можливість зручного хмарного розгортання   | 2.Нм | 2.Технологічна |
|  | 3. Можливість керування системою контролю доступом  | 3.Нм | 3.Технологічна |
|  | Якість: згідно до стандарту ISO 4444 буде проведено тестування  |      |                |
| Маркування відсутнє  |   |      |                |
| Моя компанія: "Security integration"                                   |   |      |                |
| III. Товар із підкріпленням  | 1-місячна пробна безкоштовна версія   |      |                |
|  | Постійна підтримка для користувачів   |      |                |
| За рахунок чого потенційний товар буде захищено від копіювання: патент |   |      |                |

Таблиця 5.19 — Опис трьох рівнів моделі товару

Було описано три рівні моделі товару, з чого можна зробити висновок, що основні властивості товару у реальному виконанні є нематеріальними та технологічними. Також було надано сутність та складові товару у задумці та товару з підкріпленням. Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. У даному випадку найбільш вірогідним гарантом буде патент.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проекту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (таблиця 5.20).

| <i>№ n/n</i> | <i>Рівень цін на товари-замінники, грн.</i> | <i>Рівень цін на товари-аналоги, грн.</i> | <i>Рівень доходів цільової групи споживачів, грн.</i> | <i>Верхня та нижня межі встановлення ціни на товар/послугу, грн.</i> |
|--------------|---|---|---|--|
| 1.           | 25000                                       | 20000                                     | 100000  | 21000-24000  |

Таблиця 5.20 — Визначення меж встановлення ціни

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 5.21).

| <i>№ n/n</i> | <i>Специфіка закупівельної поведінки цільових клієнтів</i>               | <i>Функції збуту, які має виконувати постачальник товару</i> | <i>Глибина каналу збуту</i>                 | <i>Оптимальна система збуту</i> |
|--------------|--|--|---|---------------------------------|
| 1.           | Придбання підписки та оплата щомісячних внесків для продовження ліцензії | Продаж   | 0(напрямую),<br>1(через одного посередника) | Власна та через посередників    |

Таблиця 5.21 — Формування системи збуту

Отже, система приносить прибуток завдяки щомісячним внескам для подовження ліцензії та придбанням підписок.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 5.22).

| <i>№<br/>п/<br/>п</i> | <i>Специфіка<br/>поведінки<br/>цільових<br/>клієнтів</i>   | <i>Канали<br/>комунікацій,<br/>якими<br/>користують<br/>ся цілові<br/>клієнти</i> | <i>Ключові<br/>позиції,<br/>обрані<br/>для<br/>позиціон<br/>ування</i> | <i>Завдання<br/>реклам<br/>ного<br/>повідомл<br/>ення</i>    | <i>Концепці<br/>я<br/>рекламно<br/>го<br/>звернення</i>               |
|-----------------------|--|---|--|--|---|
| 1.                    | Придбання ліцензії на користування в мережі Інтернет, щомісячне її продовження, користування сервісом у хмарі або ж на власних серверах. | Інтернет  | Інтеграція , хмарне розгортання  | Показати переваги сервісу, у тому числі і перед конкурентами | Демо-ролик із використання, рекламні оголошення на популярних сайтах. |

Таблиця 5.22 — Концепція маркетингових комунікацій

Отже, в таблиці 5.22 наведено концепцію маркетингових комунікацій, було визначено, що придбання ліцензії на користування буде здійснюватись в мережі Інтернет, необхідним буде щомісячне її продовження, користування сервісом можливе у хмарі або ж на власних серверах.

## 5.6. Економічне обґрунтування проекту

Згідно до проведених досліджень існує можливість ринкової комерціалізації проекту. Також, варто відмітити, що існують перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження не є високими, а проект має дві значні переваги перед конкурентами. Для успішного виконання проекту необхідно реалізувати програму із використанням засобів Biostar 2, AWS Lambda.

Для успішного виходу на ринок у продукту повинні бути наступні характеристики:

- Наявність універсального API.
- Можливість зручного хмарного розгортання.
- Можливість контролю системою контролю доступом.

В рамках даного дослідження були розраховані основні фінансово-економічні показники проекту, а також проведений менеджмент потенційних ризиків. Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

Було визначено такі сильні сторони: можливість контролю та налаштування системи контролю доступом, наявність REST API, можливість хмарного розгортання. Серед слабких сторін можна виділити можливість зміни тарифів провайдером хмарного розгортання на платні.

Можливості для виходу на ринок включають стрімкий ріст попиту на системи контролю доступом впровадження нових ризонерів, стрімке зростання росту ринку, обслуговування додаткових груп споживачів, розширення асортименту можливих послуг. Наявні такі фактори загроз: конкуренція, зміна потреб користувачів, зміна тарифів провайдера хмарного розгортання на платні, надходження на ринок альтернативних продуктів, уповільнення росту ринку.

## ВИСНОВКИ

В ході даної роботи, мною були проаналізовані існуючі системи контролю та управління доступом, виявлені їх переваги та недоліки. На підставі цих даних, була розроблена концепція власної системи з використанням хмарних технологій. Було обрано Amazon Web Services, як основного постачальника хмарних сервісів, через можливість масштабування застосунку. Проаналізувавши існуючі датчики, контролери та запірні пристрої, був складений список обладнання для реалізації системи. Після надання необхідного обладнання, була спроектована і реалізована система програмно-апаратного комплексу.

Створений програмно-апаратний комплекс вирішує основні проблеми існуючих систем:

- Розроблено прикладний програмний інтерфейс, який надає віддалений доступ до системи з будь-якої точки світу за наявності підключення до мережі інтернет.
- Створений програмно-апаратний комплекс має можливість гнучкого масштабування, оскільки розгорнутий в хмарному середовищі.
- Розроблена система не прив'язана до конкретного постачальника апаратного обладнання та має можливість розширення функціоналу.

Для масштабування системи досить закупити нове обладнання і підключити його до системи.

Згідно до проведених досліджень існує можливість ринкової комерціалізації проекту. Також, варто відмітити, що існують перспективи впровадження з огляду на потенційні групи клієнтів, бар'єри входження не є високими, а проект має переваги перед конкурентами.

Також мною були проаналізовані шляхи для вдосконалення системи, а саме:

- Реалізувати зберігання номерів карт всередині пам'яті контролерів. Дане нововведення змусить систему функціонувати в разі проблем з сервером або каналом зв'язку. Щоб в такому разі не втратити інформацію про проходи, вона

також буде записуватися в пам'ять пристрою. Після того, як зв'язок з сервером налагодиться, інформацію про проходи буде передана на сервер, а від нього буде отримана інформація про актуальність даних карт.

- Самостійно створити плату з метою прибирання непотрібних елементів схеми, тим самим збільшивши доступне для зберігання місце.
- Додати в прикладний інтерфейс можливість обмеження доступу користувачів.
- Додати в прикладний інтерфейс можливість побудови звітів по кожному із відвідувачів.
- Додати в прикладний інтерфейс можливість додавання інформації про одноразовий прохід людини без пропуску.
- Додати в систему можливість відкривання дверей без карти, наприклад, в разі відвідування кур'єром.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Галич І. В. Вибір і застосування систем контролю і управління доступом / Іван Васильович Галич. – Київ, 2010. – 169 с.
2. Скит Д. Статичне моделювання технічних систем / Джон Скит., 1993. – 164 с.
3. Фінкенцеллер К. Довідник по RFID. Теоретичні основи і практичне застосування індуктивних радіопристроїв / Клаус Фінкенцеллер., 2008. – 496 с.
4. Рикунов В. Охоронні системи та технічні засоби фізичного захисту об'єктів / Володимир Рикунов., 2008.
5. Тихонов В. Системи контролю і управління доступом / Володимир Тихонов., 2011.
6. Скородумов Б. Стандарти інформаційної безпеки / Богдан Скородумов., 2001.
7. Бахман Ф. Документування архітектури програмного забезпечення / Фелікс Бахман., 2010. – 332 с.
8. Перрі Д. Основи для вивчення архітектури програмного забезпечення / Девід Перрі., 1992.
9. Бас Л. Архітектура програмного забезпечення на практиці / Лен Бас., 2012.
10. Стаффорд Д. Документування архітектури програмного забезпечення / Джудіт Стаффорд., 2010.
11. Бабар М. Управління знаннями про архітектуру програмного забезпечення / Мухаммед Бабар., 2009.
12. Веббер Д. REST на практиці / Д. Веббер, Я. Робінсон., 2010.
13. Річардсон Л. RESTful Web Services / Леонард Річардсон., 2013.
14. Лі Г. Хмарні обчислення: принципи, системи та застосування / Гіллам Лі., 2012.
15. Кавіс М. Архітектура хмарного середовища / Майкл Кавіс., 2009.
16. Малхоленд Е. Хмарні обчислення підприємств / Енді Малхоленд., 2010.
17. Бартон Д. Офіційний навчальний посібник AWS / Джо Бартон., 2017.
18. Ерл Т. Хмарні обчислення: поняття, технології та архітектура / Томас Ерл., 2018.

19. Мартес Т. Безпека та захищеність конфіденційності у хмарі / Тім Мартес., 2010. — 320 с.
20. Шилдт Г. Java: повний посібник / Герберт Шилдт., 2017. — 615 с.
21. Бадд Т. Об'єктно-орієнтоване програмування в дії. / Томас Бадд., 2010.
22. Кушнер Д. Створення Ардуїно. / Девід Кушнер., 2017.
23. Петін О. Проекти з використанням контролера Arduino. / Олександр Петін., 2014.
24. Хаз Ф. Документація Biostar 2 / Фредерік Хаз., 2016.
25. Форд Д. Документація Suprema Xpass S2 / Деліла Форд., 2018. — 43 с.
26. Сібсанкар. Розробка і впровадження систем баз даних SQLite / Сібсанкар. — Бангалор, 2015. — 317 с.
27. Васвані. MySQL: використання і адміністрування / Васвані., 2011. — 368 с.
28. Панченко І. PostgreSQL: вчора, сьогодні, завтра. Відкриті системи. / Іван Панченко., 2010. — 213 с.
29. Кларк Д. Амазон вмикає сервіс хмарних баз даних DynamoDB / Джек Кларк., 2012.
30. Новіков А. Ефективне використання ERwin Data Modeler / Андрій Новіков., 2010. — 140 с.
31. Рейхерт Е. API тестування / Еймі Рейхерт., 2015. — 365 с.

# ДОДАТОК А

Серверна частина веб-додатку інтернет-сервісу “Відкритий спортмайданчик  
з е-сервісами”

Апробація

УКР.НТУУ “КПІ” ім. І. Сікорського. КА4218

Аркушів 7

2019

Факультет інформаційних технологій та управління  
Кафедра комп'ютерних наук і математики

# **ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ – 2019**

**Збірник тез  
VI Всеукраїнської науково-практичної конференції молодих  
науковців**

16 травня 2019 року  
м. Київ

Київ – 2019

|  |     |
|--|-----|
| Потапчук О. ....   | 95  |
| ВИКОРИСТАННЯ ХМАРНОГО СЕРВІСУ CANVA ДЛЯ РОЗВИТКУ ЦИФРОВОЇ<br>ТВОРЧОСТІ   |     |
| Прачук О.С. ....   | 97  |
| ЗАСТОСУВАННЯ МОБІЛЬНИХ ТЕХНОЛОГІЙ ПРИ ВИВЧЕННІ<br>МАТЕМАТИКИ В ЗАГАЛЬНООСВІТНІЙ ШКОЛІ  |     |
| Приткова К.В., Рикова Л.Л. ....  | 99  |
| ПЕРСПЕКТИВИ ВИКОРИСТАННЯ КОДЕКСУ ПРАКТИК<br>ПРОТИ ДЕЗІНФОРМАЦІЇ У КОНТЕКСТІ ЗАГАЛЬНОДЕРЖАВНИХ<br>ВИБОРІВ В УКРАЇНІ               |     |
| Прокопов В.В. ....   | 101 |
| ВИКОРИСТАННЯ ІКТ ПРИ ПІДГОТОВЦІ МАЙБУТНІХ ФАХІВЦІВ В<br>ДОННАБА  |     |
| Пучков І.Р., Єщенко М.Г. ....  | 103 |
| ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У ПРОЦЕСІ РОБОТИ З<br>НАВЧАЛЬНОЮ ТЕРМІНОЛОГІЄЮ   |     |
| Рикова Л. Л., Медведська О. П. ....  | 104 |
| СЕРВЕРНА ЧАСТИНА ВЕБ-ДОДАТКУ ІНТЕРНЕТ СЕРВІСУ «ВІДКРИТИЙ<br>СПОРТ-МАЙДАНЧИК З Е-СЕРВІСАМИ»                                       |     |
| Сербін А.В., Ковальчук А. М. ....  | 106 |
| ПЕРЕВАГИ І ОСОБЛИВОСТІ ВИКОРИСТАННЯ ВІЛЬНИХ<br>КРОССПЛАТФОРМОВИХ СЕРЕДОВИЩ ПРОГРАМУВАННЯ   |     |
| Тарасенко Є.О. ....  | 107 |
| МЕТОДИЧНІ ЗАСАДИ РЕАЛІЗАЦІЇ ПРОЦЕСУ КОМПЛЕКСУВАННЯ<br>ПІД ЧАС МОДУЛЬНОГО ПРОГРАМУВАННЯ   |     |
| Ткаченко О.С. ....   | 108 |
| AR-ТЕХНОЛОГІЇ В ОСВІТНЬОМУ ПРОЦЕСІ   |     |
| Чернецька Ю. М. ....   | 110 |
| ОСОБЛИВОСТІ РОЗВИТКУ ІНФОРМАЦІЙНО-ДОСЛІДНИЦЬКОЇ<br>КОМПЕТЕНТНОСТІ АСПІРАНТІВ І ДОКТОРАНТІВ                                       |     |
| Яцишин А.В. ....   | 113 |
| ГЕОІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ЯК ЗАСІБ ФОРМУВАННЯ ЕКОЛОГІЧНОЇ<br>КОМПЕТЕНТНОСТІ МАЙБУТНІХ ДОКТОРІВ ФІЛОСОФІЇ З КОМП'ЮТЕРНИХ<br>НАУК |     |

**СЕРВЕРНА ЧАСТИНА ВЕБ-ДОДАТКУ  
ІНТЕРНЕТ СЕРВІСУ  
«ВІДКРИТИЙ СПОРТ-МАЙДАНЧИК З Е-СЕРВІСАМИ»**

Сербін А.В., Ковальчук А. М.

*Національний технічний університет України "Київський політехнічний інститут  
імені Ігоря Сікорського", м. Київ*

Дозвілля як частина вільного часу, залучає молодь своєю нерегламентованістю і добровільністю вибору його різних форм, демократичністю, емоційною насиченістю, можливістю поєднати в ньому фізичну й інтелектуальну діяльність, творчу і споглядальну, виробничу й ігрову. Для значної частини молодих людей соціальні інститути дозвілля є основними джерелами соціально-культурної інтеграції й особистої самореалізації. Однак усі ці переваги діяльності сфери дозвілля поки ще не стали надбанням, звичним атрибутом способу життя усієї молоді. В 21 столітті існує проблема популяризації спорту серед молоді. Наявно безліч причин появи цієї проблеми, починаючи від нерозуміння людьми важливості спорту, активного відпочинку, закінчуючи тим, що в Україні не розвинена інфраструктура для занять спортом (наприклад, спортивні майданчики).

Отже, проблема полягає в тому, щоб надати людям зручний спосіб бронювання та планування заходів на спортивному майданчику, який розташований на базі кампусу КПІ, а також створити надійну систему пропуску до спортивного майданчику.

Тому актуальною є розробка програмного забезпечення яке надасть користувачам можливість бронювання заходів, купування квитків, а адміністраторам майданчику – змогу контролю доступу до спортивного майданчику.

Система ділиться на три частини. Перша частина – це турнікети та програмне забезпечення, яке працює з ними (STOP-NET). Система

107

контролю доступу (СКД) STOP-Net 4.0 реалізована на платформі інтегрованої системи безпеки (ІСБ) STOP-Net 4.0 і є самостійним продуктом, призначеним для вирішення завдань організації і управління фізичним доступом співробітників і відвідувачів на територію і в окремі приміщення об'єкту. Друга частина – це front-end та android додатки, з якими працює користувач. Третя частина – це серверна частина. Вона об'єднує систему контролю доступу STOP-NET та front-end і android додатки. Вона буде реалізована за допомогою мови програмування Java, з використанням таких фреймворків як Spring MVC, тощо. Також в її обов'язки входить робота з базою даних розкладу заходів, робота з базою даних особистої інформації користувачів, генерація електронного квитка, тощо.

На даний момент існують системи, які реалізують схожий функціонал. Мінус цих систем – це неможливість роботи з студентами та майданчиками які знаходяться на базі кампусу КПІ.

Отже, розроблювана система дозволить користувачеві бути проінформованим про спортивні активності, які будуть проходити в університеті, а також надасть йому зручний спосіб використання спортивної інфраструктури КПІ.

#### ДЖЕРЕЛА

1. Захаріна С. А. Організаційні умови вдосконалення фізичного виховання у вищому закладі освіти / Євгенія Анатоліївна Захаріна., 2007.
2. Бальсевич В. К. Інтелектуальний вектор фізичної культури людини / Вадим Константинович Бальсевич., 1991.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

# СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVII Міжнародної  
науково-практичної конференції  
молодих вчених та студентів  
м. Київ, 23-26 квітня 2019 року,

ТОМ 2



Київ- 2019

|   |     |
|---|-----|
| <b>Використання методів виявлення автоматичних моделей поведінки для побудови аналітичних сценаріїв.</b>              | 122 |
| <i>ОЛЄНСЬКА К.М., аспірант;</i>   |     |
| <i>ШПУРИК В.В.</i>  |     |
| <i>Керівник - доц., к.т.н. Коваль О.В.</i>  |     |
| <b>Візуалізації структури KNX-мережі з використанням людинно-машинного інтерфейсу.</b>                                | 123 |
| <i>ЯШИН А.С., магістрант гр. ТМ-81мп</i>  |     |
| <i>Керівник - доц., к.е.н. Сеґеда І.В.</i>  |     |
| <b>Розробка агента моніторингу і управління попитом на електричну енергію "Розумної будівлі".</b>                     | 124 |
| <i>ШАРНІН С.А., магістрант гр. ПІ-81мп</i>  |     |
| <i>Керівник - доц., к.т.н. Ковальчук А.М.</i>   |     |
| <b>Агент моніторингу та управління режимами роботи мікроенергостанцій.</b>  | 125 |
| <i>СТОЛЯР А.В., магістрант гр. ТВ-81мп</i>  |     |
| <i>Керівник - доц., к.т.н. Ковальчук А.М.</i>   |     |
| <b>Розробка серверної частини для веб-додатку відкритий спортивний майданчик з е-сервісами.</b>                       | 126 |
| <i>СЕРБІН А.В., магістрант гр. ТВ-82</i>  |     |
| <i>Керівник - доц., к.т.н. Ковальчук А.М.</i>   |     |
| <b>Розв'язок задачі обрахунку водонагрівача в інтерактивному режимі з використанням клієнт-серверної архітектури.</b> | 127 |
| <i>РОМАНОВ О.В., магістрант гр. ТВ-61мп</i>   |     |
| <i>Керівник - доц., к.т.н. Кузьменко І.М.</i>   |     |
| <b>Оцінка територій для побудови вітроелектростанцій із застосуванням мультиагентних технологій та ГІС.</b>           | 128 |
| <i>ПІДВИШЕННИЙ Т.О., магістрант гр. ТВ-81мп</i>   |     |
| <i>Керівник - асист. Швайко В.Г.</i>  |     |
| <b>REST-інтерфейс як основа комунікації систем контролю доступу.</b>  | 129 |
| <i>ОБРУСНИК Д.В., студент гр. ТВ-82мп</i>   |     |
| <i>Керівник - доц., к.е.н. Левченко Л.О.</i>  |     |
| <b>Інструментальні засоби аналізу впливу параметрів експерименту на сигнатуру морського об'єкту.</b>                  | 130 |
| <i>ОБІЩЕНКО А.А., магістрант гр. ТВ-81мп</i>  |     |
| <i>Керівник - доц., к.т.н. Варава І.А.</i>  |     |
| <b>Система моделювання структури та функціонального контенту інженерних систем енергоефективної будівлі.</b>          | 131 |
| <i>КУРСЕНКО Л.О., магістрант гр. ПІ-81мп</i>  |     |
| <i>Керівник - доц., к.т.н. Шпурик В.В.</i>  |     |
| <b>Інструментальний засіб підтримки динамічного реєстру інформаційних ресурсів на базі ОРБД Caché.</b>                | 132 |
| <i>КОСТЕНКО О.П., магістрант гр. ТВ-81мп</i>  |     |
| <i>Керівник - доц., к.т.н. Кублій Л.І.</i>  |     |
| <b>Мультиагентні системи в децентралізованих мережах енергоспоживання.</b>  | 133 |
| <i>ЖОРНОВИЙ Е.Г., магістрант гр. ТВ-82</i>  |     |
| <i>Керівник - доц., к.т.н. Ковальчук А.М.</i>   |     |
| <b>Розробка веб-додатку для відкритого спортивного майданчику з е-сервісами.</b>                                      | 134 |
| <i>АМБРОС С.М., магістрант гр. ТВ-81мп</i>  |     |
| <i>Керівник - доц., к.т.н. Ковальчук А.М.</i>   |     |

УДК 621.43.056:632.15

Магістрант 5 курсу, гр. ТВ-82 Сербін А.В.

Доц., к.т.н. Ковальчук А.М.

## **РОЗРОБКА СЕРВЕРНОЇ ЧАСТИНИ ДЛЯ ВЕБ-ДОДАТКУ ВІДКРИТИЙ СПОРТИВНИЙ МАЙДАНЧИК З Е-СЕРВІСАМИ**

Дозвілля як частина вільного часу, залучає молодь своєю нерегламентованістю і добровільністю вибору його різних форм, демократичністю, емоційною насиченістю, можливістю поєднати в ньому фізичну й інтелектуальну діяльність, творчу і споглядальну, виробничу й ігрову. Для значної частини молодих людей соціальні інститути дозвілля є основними джерелами соціально-культурної інтеграції й особистішої самореалізації. Однак усі ці переваги діяльності сфери дозвілля поки ще не стали надбанням, звичним атрибутом способу життя усієї молоді. В 21 столітті існує проблема популяризації спорту серед молоді. Наявно безліч причин появи цієї проблеми, починаючи від нерозуміння людьми важливості спорту, активного відпочинку, закінчуючи тим, що в Україні не розвинена інфраструктура для занять спортом (наприклад, спортивні майданчики).

Отже, проблема полягає в тому, щоб надати людям зручний спосіб бронювання та планування заходів на спортивному майданчику, який розташований на базі кампусу КПІ, а також створити надійну систему пропуску до спортивного майданчику.

Тому актуальною є розробка програмного забезпечення яке надасть користувачам можливість бронювання заходів, купування квитків, а адміністраторам майданчику – змогу контролю доступу до спортивного майданчику.

Система ділиться на три частини. Перша частина – це турнікети та програмне забезпечення, яке працює з ними (STOP-NET). Система контролю доступу (СКД) STOP-Net 4.0 реалізована на платформі інтегрованої системи безпеки (ІСБ) STOP-Net 4.0 і є самостійним продуктом, призначеним для вирішення завдань організації і управління фізичним доступом співробітників і відвідувачів на територію і в окремі приміщення об'єкту. Друга частина – це front-end та android додатки, з якими працює користувач. Третя частина – це серверна частина. Вона об'єднує систему контролю доступу STOP-NET та front-end і android додатки. Вона буде реалізована за допомогою мови програмування Java, з використанням таких фреймворків як Spring MVC, тощо. Також в її обов'язки входить робота з базою даних розкладу заходів, робота з базою даних особистої інформації користувачів, генерація електронного квитка, тощо.

На даний момент існують системи, які реалізують схожий функціонал. Мінус цих систем – це неможливість роботи з студентами та майданчиками які знаходяться на базі кампусу КПІ.

Отже, розроблювана система дозволить користувачеві бути проінформованим про спортивні активності, які будуть проходити в університеті, а також надасть йому зручний спосіб використання спортивної інфраструктури КПІ.

Перелік посилань:

1. Захаріна Е. Організаційні умови вдосконалення фізичного виховання у вищому закладі освіти / Захаріна Е. – 2007.
2. Бальсевич В. К. Інтелектуальний вектор фізичної культури людини / Вадим Бальсевич. – 1991.