Khmelnytskyi National University
Faculty of Information Technologies
Department of computer engineering and information systems

QUALIFICATION WORK

bachelor
Educational level

Software and technical tool for increasing the security of the cyber-physical system "Smart parking"
Topic name

QWCE 19004.19.01.02 EN
Cipher

Field of knowledge 12 "Information technology"
Cipher, name

Specialty 123 "Computer Engineering"
Cipher, name

Educational program " Computer Engineering and Programming"
Name

Performed by: IV year student, group КІін–19–1 _____ R. Kawonga
Signature — Initials, surname

Supervisor _____ O.O. Pavlova
Signature, date — Initials, surname

Normocontroller _____ S.M. Lysenko
Signature, date — Initials, surname

Approved by:
Head of Computer engineering and information systems department _____ T.O. Hovorushchenko
Signature — Initials, surname

«5» June 2023 p.

Khmelnytskyi 2023

KHMELNYTSKYI NATIONAL UNIVERSITY

Faculty of INFORMATION TECHNOLOGIES

Department OF COMPUTER ENGINEERING AND INFORMATION SYSTEMS

Bachelor's degree

Field of knowledge 12 INFORMATION TECHNOLOGY

Specialty 123 COMPUTER ENGINEERING

Educational program EDUCATIONAL PROGRAM "COMPUTER ENGINEERING AND PROGRAMMING"

APPROVED by

Head. of department

T.O.Hovorushchenko

" 11 " 01 202

TASK
FOR THE QUALIFICATION BACHELOR WORK

Kawonga Rose

Surname, name, patronymic of the student

1. Project topic (work): Software and Technical Tool for increasing the Security of Cyber-Phys System "Smart Parking"

Project Manager (work) Pavlova O.O., PhD, Senior lecturer.

Surname, name, patronymic, degree, academic rank

Approved by the order of the rector of the university from 01.03.2023 №5

2. The deadline for student submission of the project (work) to the department 09.06.2023

3. Initial data for the project (work) Diploma design tasks

4. The content of the explanatory note (list of issues to be developed)

Review of existing systems for solving the problem

Security problem of cyberphysical smart parking system

Implementation of software and hardware to improve the security of the cyber-physical syst "Smart Parking"

5. List of graphic material (indicating mandatory drawings)

Structural diagram of a cyber-physical system for smart parking, using middleware

Scheme of operation of the components of the software and technical tool for increasing security of the cyber-physical system of smart parking

Cyber-Physical System Architecture for Smart Parking Using Middleware

6. Consultants sections of the diploma project (work)

Section	Surname, initials and position of the consultant	Signature, date	
		Task issued	Task accepted
Normocontrol	Lysenko S.M., Professor of CEIS department		
Anti-plagiarism	Nicheporuk A.O., Associate Professor CEIS department		

7. Date of issue of the task «_11_»_____01_____2023 p.

CALENDAR PLAN

Nos/n	Name of stages (sections) diploma project (work)	Deadline stages of the project (work)	Note
1	Choosing the direction of research and coordinating the topics of qualification work with the head	11.01.2023	Performed
2	Familiarization with the subject area; formulation of the purpose and objectives of the study; definition of the object and subject of research	01.02.2023	Performed
3	Work on Chapter 1 – Overview of existing systems for solving the problem	01.03.2023	Performed
4	Work on Chapter 2 – The problem of Smart Parking Cyber-Physical System Security	01.04.2023	Performed
5	Work on section 3 – Implementation of Software and Technical Tool for increasing the Security of Smart Parking Cyber-Physical System	30.04.2023	Performed
6	Execution of an explanatory note according to the requirements	21.05.2023	Performed
7	Preliminary defence of WRC	26.05.2023	Performed
8	Defence of the WRC at the EC meeting	June 2023	

Student _____ (Signature) R. Kawonga (Initials, surname)

Supervisor _____ (Signature) O. O. Pavlova (Initials, surname)

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Text documents			
1		QWCE 19004.19.01.02 EN	Explanatory note	65		
			Graphic materials			
2		QWCE 19004.04.01.02 E8	Structural diagram of a cyber-physical system for smart parking, using middleware	1		
3		QWCE 19004.04.01.02 E8	Scheme of operation of the components of the software and technical tool for increasing the security of the cyber-physical system of smart parking	1		
4		QWCE 19004.04.01.02 E8	Cyber-Physical System Architecture for Smart Parking Using Middleware	1		

Зм	Арк	№ докум	Підпис	Дата	QWCE. 19004.19.01.02 EN		
Розробив		Kawonga			Відомість проекту	Літера	Аркуш 1 / Аркушів 1
Перевір.		Pavlova				У	
Н. контр.		Lysenko				ХНУ, Кіін-19-1	
Затв.		Hovorushchenko		05.06			

ABSTRACT

Theme of qualification work: Software and hardware to improve the security of the cyber-physical system "Smart parking".

Author: Kawonga Rose.

Supervisor: Pavlova Olga.

Explanatory note: 65 pp., 15 fig., 5 tab., 3 app., 52 sources.

Graphic part: 3 posters.

SMART PARKING, CYBERPHYSICAL SYSTEM, SOFTWARE AND HARDWARE, OUTDOOR SURVEILLANCE CAMERA, SECURITY, CYBER PHYSICAL SYSTEM ARCHITECTURE

The purpose of the qualification work is to increase the security of the cyber-physical system "Smart parking".

The object of research is the process of improving the security of the cyber-physical system "Smart parking".

The subject of the study is software and hardware to improve the security of the cyber-physical system "Smart parking".

To achieve this goal, various research methods are used, such as synthesis, analysis and modeling of processes, principles of system analysis, as well as approaches based on set theory.

Student Signature

05.06.2023

Date

CONTENT

INTRODUCTION 5

1 OVERVIEW OF EXISTING SYSTEMS FOR SOLVING THE PROBLEM 7

1.1 Software and technical tools for smart parking in Ukraine and the world 7

1.2 Comparison of the existing software and technical tools 16

1.3 Analysis of custom applications for smart parking 19

1.4 Conclusions 23

2.1 Client-Server Architecture and Possible Security Threat Factors 24

2.1.1 Client-side security risks 25

2.1.2 Server-side security risks 29

2.2 API Security Risks 30

2.3 Selection of methods and environment for software implementation 32

2.4 Inconclusions 36

3 IMPLEMENTATION OF SOFTWARE AND TECHNICAL TOOL FOR INCREASING THE SECURITY OF SMART PARKING CYBER-PHYSICAL SYSTEM 38

3.1 The proposed method of increasing the Security of Smart Parking Cyber-Physical System 38

3.2 Structural diagram and algorithm of operation of Software and Technical Tool for increasing the Security of Smart Parking Cyber-Physical System 43

3.3 Conclusions 62

CONCLUSIONS 63

LIST OF REFERENCE SOURCES 64

APPENDIX A Structural diagram of a cyber-physical system for smart parking, using middleware 70

APPENDIX B Scheme of operation of the components of the software and technical tool for increasing the security of the cyber-physical system of smart parking 73
APPENDIX C Cyber-physical system architecture for smart parking using middleware..... 74

ABBREVIATIONS AND CONDITIONAL SIGNS

SS – software system

OS – operating system

HTTP – HyperText Transfer Protocol – Hypertext Data Transfer Protocol

HTTPS – HyperText Transfer Protocol Secure – Secure Hypertext Data Transfer Protocol

RFID – Radio Frequency Identification – electromagnetic sensor

API – Applied Program Interface

INTRODUCTION

At the present stage of development of information and computer technologies, special attention should be paid to security issues in software development. This is especially important for critical software and software of cyber-physical systems, as data loss or malfunctions can have unpredictable and sometimes critical consequences. Incorrect operation of the algorithm or errors in image recognition by an artificial neural network can lead to an incorrect result. Since the client-server architecture is particularly vulnerable to various kinds of external threats, it is advisable to provide methods and algorithms for the protection and security of the smart parking system in the early stages of the life cycle, that is, at the design stage of the software architecture. This is extremely important because the cost of bug fixing increases with each stage of the life cycle.

Computer equipment (such as smartphones, wireless sensors and personal computers) is becoming smaller, cheaper and more powerful as the technical basis of smart parking systems. As a result, mobile and ubiquitous computing are rapidly becoming important components of dispersed network computing infrastructures. It offers us a powerful platform for computing real-time information from the physical world (physical component) and communicating with people (cyber part). We should not underestimate the importance of cyber-physical systems in everyday life.

The relevance of the work lies in the development of a subsystem to ensure the security of the cyber-physical system of smart parking. The purpose of the thesis is to increase the security of the cyber-physical system for smart parking.

The goal is achieved by solving the following main tasks:

1) review existing solutions and security systems for smart parking;

2) select components and environment for the task;

3) develop a subsystem to ensure the security of the cyber-physical system for smart parking.

The object of research is the process of developing a subsystem to ensure the security of the cyber-physical system for smart parking. The subject of the study is a subsystem for ensuring the security of the cyber-physical system for smart parking.

The practical value of the obtained results lies in the development of a subsystem for ensuring the security of the cyber-physical system for smart parking.

On the topic of the thesis, she took part in the All-Ukrainian Scientific and Practical Conference Information Technologies and Engineering (IT&I-2023), Mykolaiv and published abstracts in the collections of the conference.

1 OVERVIEW OF EXISTING SYSTEMS FOR SOLVING THE PROBLEM

1.1 Software and technical tools for smart parking in Ukraine and the world

The problem with free parking spaces exists both in Ukraine and in many other countries of the world. The main problems associated with parking were considered. The results of the analysis are shown in Table 1.1.

Table 1.1 – Results of the analysis of problems with free parking spaces in Ukraine and the world

Problem	Description
Insufficient number of parking spaces	Many cities and towns do not have enough parking spaces to meet the needs of residents, visitors and businesses. This leads to drivers searching for free spaces and parking on sidewalks, lawns or in prohibited places.
Unauthorized parking	Many drivers park their cars in prohibited places such as pedestrian crossings, yard exits, emergency driveways, roadsides, etc. This creates problems for road safety, restricts access to other drivers and interferes with the normal functioning of the city.

End of the table 1.1 – Results of the analysis of problems with vacant parking spaces in Ukraine and the world

High cost of parking	In some cities, parking can be very expensive, especially in central areas or in commercial areas. This can be financially demanding for drivers and creates inequality in access to parking spaces.
Inefficient use of available spaces	It often turns out that existing parking spaces are not used efficiently. For example, some cars may occupy seats for long periods of time even when not in use, while other drivers cannot find empty seats.
Insufficient infrastructure	The lack of adequate parking infrastructure, such as parking near public buildings, shopping malls, public transport stations, etc., makes it difficult for drivers to find parking spaces.

To address these concerns, governments and municipalities can take measures such as:

1. Expansion of parking infrastructure through the construction of new parking lots and multi-storey parking lots.

2. Introduction of electronic parking management systems that allow drivers to find free spaces and pay for parking using mobile applications or the Internet.

3. The use of "smart parking" technologies that use sensors and control systems to optimize the use of parking spaces and improve parking efficiency.

4. Legislative measures that provide for the introduction of fines for parking in prohibited places and improper parking.

5. Promoting the use of alternative modes of transport, such as bicycles, electric transport, public transport, which will reduce the need for parking spaces for cars.

Understanding these problems and taking appropriate measures can contribute to improving the parking situation both in Ukraine and in other countries.

A study was conducted to solve the problem of smart parking system security using various methodologies and tools. In terms of providing a Smart Parking software security system, the main criteria that must be met when creating a security system for smart parking are hardware security, hardware and software security, and software security.

The following requirements can be distinguished:

1. checking the parameters of secure access to the database;
2. security of the client program;
3. server security;
4. API security if the design of the smart parking software system supports its use;

The basic principle of smart parking systems is that the decision to disseminate traffic information should be made by infrastructure, and not by individual cars, which may have inaccurate or incomplete route information. The infrastructure in such a system is achieved by installing sensor belts at regular intervals on the road. Each belt consists of several piezoelectric pressure detectors, a rudimentary unit and a fusion

engine, as well as several small receivers. The pressure sensors in each belt allow each message to be linked to a real vehicle passing through the belt, eliminating the need for individual vehicle identification, avoiding safety concerns. There are two immediate benefits of implementing belts instead of roadside infrastructure. Firstly, the belts are much less vulnerable to manipulation, and secondly, they are better positioned to detect passing cars and grip them in a simple and safe way. Vehicles are equipped with a Tamper Protection (TRD) device. TRD receives information from all components of the car, including the wireless transceiver unit, speedometer, gas tank readings, tire pressure sensors and ambient temperature sensors. It is worth noting that Smart Parking is an intelligent computer parking application and a revolutionary infrastructure that provides security and privacy. For starters, motorists on the road can see and reserve a parking space. Parking can be an efficient and permanent service. Secondly, by using the infrastructure of such smart parking, driver privacy is taken into account and protected. Finally, information security is ensured using belt infrastructure and encryption/decryption techniques. The simulation results show that the proposed approach leads to high parking space utilization and short time to search for a parking space. The cyber-physical system consists of four modules:

The driver module manages communication with hardware devices. The belt sensor driver, alarm driver for the vehicle's short-range transceiver and IFD driver for vehicle identification are all part of the driver module.

Communication module; receives and sends messages between the sender and the recipient. This module simulates the communication process and performs error control, such as checksum checking and error correction, to communicate the vehicle with infrastructure (V2I). Since communication response times are limited, the goal is to improve communication speed and message accuracy. This module transmits messages between two fixed transceivers for communication between infrastructures (I2I), such as a transceiver in a parking lot and a transceiver in a booth.

Functional module; this is the main function of the parking system, which includes monitoring, registration, booking and advertising management. The

functional module can communicate with hardware devices and transmit/receive data without knowing the intricacies of the lower levels thanks to its subscription to the driver module and the communication module.

Application module; controls the entire parking system. The main functions of the application include account administration (cash and credit/debit management), transaction management, fault tolerance, and service management.

The decomposition of elements of a cyberphysical system based on sensors is shown in Figure 1.1.

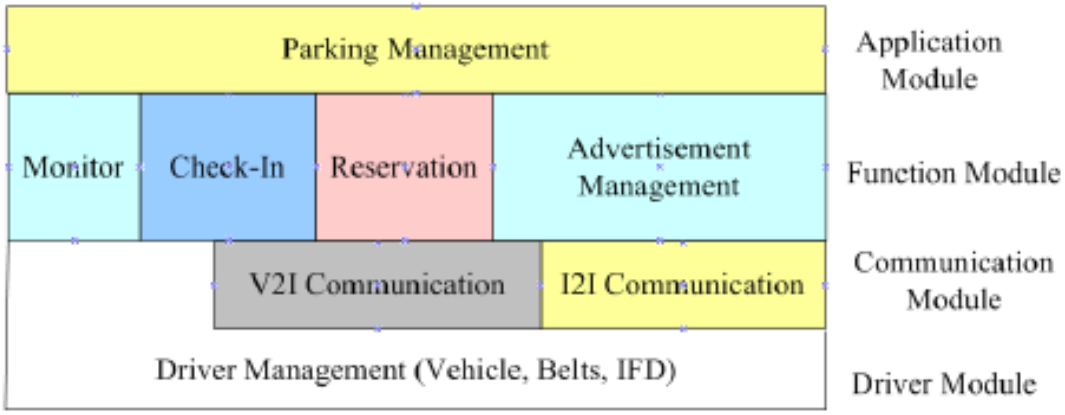


Figure 1.1 – Decomposition of elements of the cyberphysical system for smart parking based on sensors

Also today, a very popular technology for ensuring the security of smart parking is the technology of using a blockchain - blockchain.

The parking provider, blockchain network and user are the three actors in the proposed system. In the integrated system, the parking service provider provides parking as a service, updates the parking space. The blockchain network includes a public ledger that is updated only with valid transactions. A consensus process is used to confirm transactions. The participant looking for a parking space is called the parking user. To communicate with the integrated smart parking system, each participant has its own program interface.

Suppose there are many smart car parking options from different parking service companies in a city. Also suppose that for simplification, each smart parking is managed by a single service provider. Each parking lot is linked to a blockchain-based smart parking system. In general, each parking space contains a local copy of the workbook (i.e., a local block). There are two types of transactions in the system. First you need to consider the data of the parking sensor. Suppose that each smart car parking space has an Internet of Things (IoT) device (such as a parking sensor) that can generate car parking availability as a transaction. This system provides an accurate, efficient, and dependable means of automobile parking for urban cities.

The SAVP (smart autonomous vehicle parking) system is an intriguing and significant contribution to the field of smart parking. Fog-blockchain technologies were used to augment the existing IoT-cloud platform for AVs, and a proof-of-concept implementation was carried out. The system is composed of three general layers: the perception layer, the intermediate fog layer, and the cloud layer.

To complete the transaction, each car parking service provider has a smart contract. When the parking space status changes from "free" to "busy", the corresponding IoT device creates a transaction. Similarly, when a parking space changes from "busy" to "free", an IoT device creates a transaction. The transaction is first sent to the local block. The transaction is sent to the blockchain network for validation by the local block. Secondly, there is information about the cost of parking. Suppose parking service providers determine parking rates based on time. They develop smart contracts for parking fees. Wireless magnetic sensors detect the presence of a car by changes in the magnetic field, while ultrasonic sensors detect the presence of a vehicle using sound waves. The most common types of parking sensors are ultrasonic and electromagnetic.

The parking pricing smart contract is sent to the blockchain network. A transaction is created whenever the parking price changes dynamically based on time. The transaction is transmitted to the blockchain network for verification. The transaction is then verified by the blockchain network using a consensus method. If the

transaction is correct, it is recorded in the public ledger. As a result, all local blocks are updated. The decomposition of elements of the cyber-physical system for smart parking based on blockchain technology is shown in Figure 1.1.

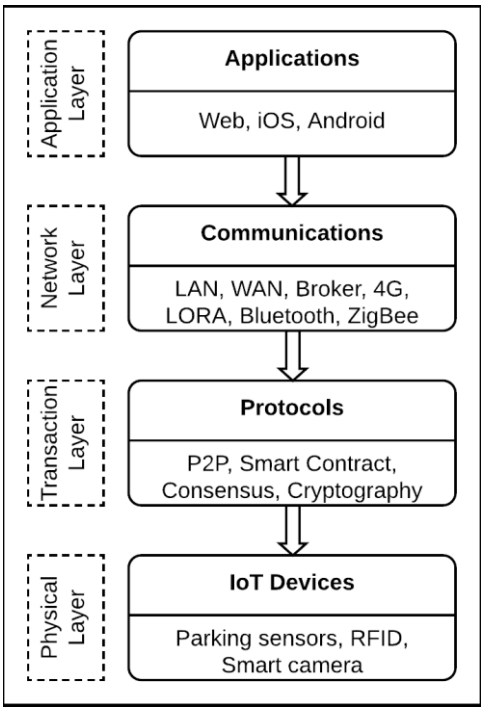


Figure 1.2 – Decomposition of elements of the cyberphysical system for smart parking based on blockchain technology

The upper level of the architectural stack is the application layer, which allows participants to interact with the system. Users can search and book parking spaces using a mobile phone app (such as Android or iOS) or a web app. Similarly, parking providers can submit park-related information (e.g. availability and offers) to the integrated system. The user connects to the blockchain network through the application layer and can use the application to send requests to the integrated parking system. The integrated system is responsible for recommending an appropriate parking zone based on user preferences and availability. Because users interact directly with the integrated system, this layer provides end users with the highest level of service.

The network layer provides communication between parking facilities, the integrated system and users. This layer will send data from users and parking locations

to the integrated system. This layer will include various communication technologies such as LAN and WAN that will be used by users, parking service providers, and parking system-related IoT devices (such as parking sensors and security cameras). As part of the standard offering, the network layer seamlessly provides distributed public ledger and content services to stakeholders. It includes many wireless technologies (such as Lora, Bluetooth, Wi-Fi, etc.) as well as modern GSM technologies such as 4G and 5G. This level also provides scalability. For example, it allows you to dynamically add and remove stakeholders from the integrated system. The network layer also ensures the security of the physical layer of the system.

The transaction layer is responsible for transactions between network nodes. It will also ensure a complete blockchain network consensus process. Users and parking facilities will securely share data through smart contracts and consensus mechanisms. Thanks to this layer, the parking center will also update the public book. This layer connects to the main blockchain network through the interface of the integrated system. This level also validates new transactions. In addition, the transaction layer maintains transaction transparency and protects data transfers in the absence of a trusted third party. We can eliminate system bottlenecks and central sources of failure with P2P-based distributed design. In addition, the blockchain will consider the transfer of user data as transactions that will be confirmed by smart contracts. Thus, user information will be immutable and distributed over time using cryptographic blockchain technology.

The physical layer consists of several kinds of IoT devices. The peer-to-peer protocol connects all these devices into a single network. The main component of this level includes various types of sensors and actuators. There will be additional built-in technologies such as the Raspberry Pi and Arduino in addition to WSN devices. The transaction layer will transport data from the IoT device to the parking center server. The peer-to-peer will then connect to the parking center servers and update the public ledger. In addition, this layer ensures traceability of sensor and drive data and responsibility throughout the peer-to-peer network. Since reliability is an important characteristic of our proposed system, data can be safely and securely transported from

IoT devices using blockchain, secure immutable storage. The availability of a particular parking space will be determined from the physical layer using sensors of IoT devices. The user will be verified using cryptography and the public ledger will be updated with information about available parking spaces. The smart contract will handle cryptographic verification techniques at the transaction level. To reserve a parking space, the user can submit a request from the application layer, which will be routed through the network layer.

Smart Car Parking technology based on infrared sensors allows you to park vehicles vertically, floor by floor, minimizing the necessary space. The system is controlled by software created with the Arduino controller, minimizing the time a person spends searching for a parking location and parking the car manually. The decomposition of elements of the cyber-physical system for smart parking based on infrared sensors is shown in Figure 1.3

An IR sensor is allocated for each parking space. The IR sensor no longer broadcasts the signal of free space after the car is parked, and the system knows which parking spaces are full. In the parking system, a CO2 sensor is used to monitor and monitor carbon dioxide. The CO2 sensor is used in the parking system to control and measure carbon dioxide. When darkness falls, LDR provides a pop-up message. The LCD display at the entrance will regularly poll all IR sensors, as well as when the car is parked and not parked, to display available spaces

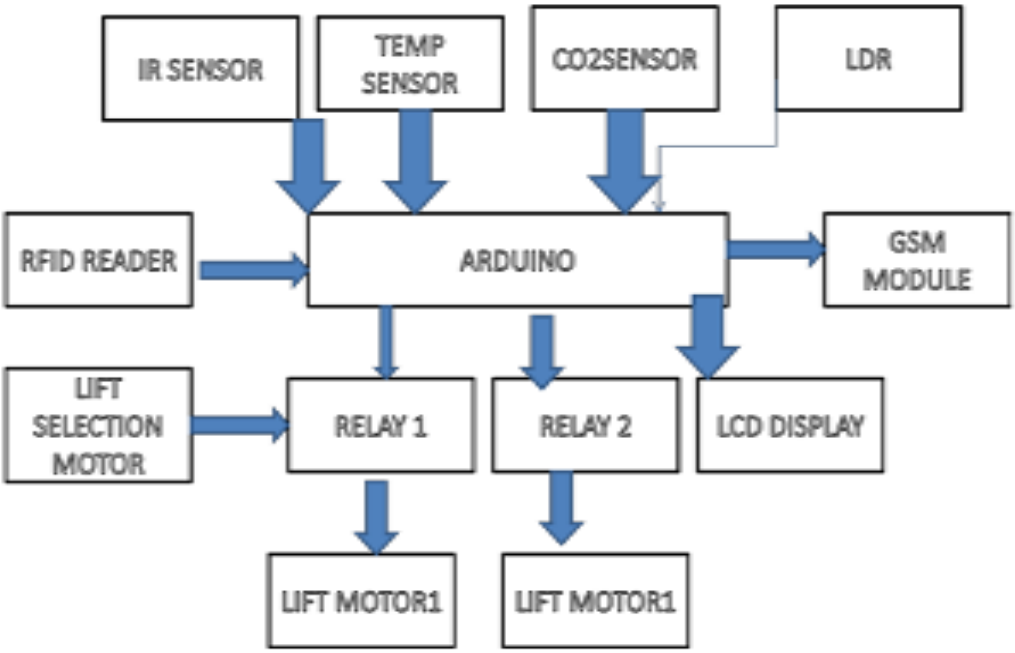


Figure 1.3 – Decomposition of elements of the cyber-physical system for smart parking based on infrared sensors

When the user detects any empty space on the LCD, they go to the RFID tag reader sensor and offer the sensor their RFID card.

The user then exits the vehicle and mounts it on an input conveyor belt. The car is then placed in a pre-specified parking space, and the number on the LCD increases, as does the real-time clock for that parking space. Now the elevator will return to ground level or neutral to meet the needs of others. RFID CARD to debit money from the user's bank account. The elevator then transports the vehicle to the ground floor, where the user can take it out of the car park. The system also includes a controller model GSM SIM 900, which will respond to the user's SMS request sent to the GSM number regarding available parking spaces by identifying free parking spaces using the Arduino controller and IR sensors installed in parking spaces. "Available parking spaces 2, 3 at 10:30 am 20.04.2017" or "No parking spaces available at this time" is an example of a notification that indicates available parking spaces. As a result, the system with all these features provides an accurate, efficient and reliable means of parking for modern cities.

1.2 Comparison of the existing software and technical tools

Numerous studies have been conducted to solve the security problem of the smart parking system using various methods and tools. The main criteria that must be guided when developing a security system for smart parking are hardware security, hardware and software security of connection and software security. From the point of view of ensuring the security of the Smart Parking software system, the following criteria can be distinguished:

1. checking the parameters of secure access to the database;
2. security of the client program;
3. server security;

Security API, if its use is provided by the architecture of the smart parking software system.

We will conduct a modern analysis of known solutions and methods aimed at improving safety. The article [6] proposes a secure smart parking system using blockchain technology, which uses masking techniques to protect the location of drivers.

In [8], a solution is provided to prevent vehicle theft in the parking lot using RFID and GSM technology. Energy saving methods based on peripheral computing and IoT are proposed in [9].

[10] presents an inclusive, long-term, efficient and well-functioning Smart Autonomous Vehicle Parking System (SAVP). The authors present an integrated smart parking system that brings together several parking service providers within a single platform with the aim of providing uniform parking information services for smart city passengers.

The main role of the study in [11] is to analyze smart parking solutions from a technical perspective, emphasizing the available systems and sensors, as noted in the literature. The review aims to provide comprehensive information on creating smart

parking solutions. A holistic study of the current state of smart parking systems should include the classification of systems such as large car detection technologies.

The article [12] proposes a parking management system focused on business entities. The proposed system will focus on privacy for different entities using the system. The article is aimed at improving existing research on smart parking using blockchain. This document proposes a parking management system that will be based on JPMorgan Quorum.

Article [13] aims to develop a secure and intelligent parking monitoring management system (SPMS) based on WSN, RFID and IoT integration.

Inspired by Blockchain technology and AI, the authors[14] propose a secure blockchain-enabled framework for energy-efficient smart parking in an environmentally safe urban environment.

JMU Secure Smart Parking through the cloud environment is proposed in [15]. Using a radio frequency identification scanner, our system can count the number of vehicles entering and leaving each parking lot on campus.

Intelligent parking system in the city based on mobile networks of the 5th generation (5G) proposed in [16]. 5G mobile technology has two important advantages: high data rates and low transmission latency, so it can better meet the rapid development of the Internet of Things (IoT).

Article [19] presents a program of work in progress, which contributes to the creation of new business solutions and the results of cutting-edge research. The authors reveal the multilevel system of the PSP business model through interdisciplinary research blocks, where original results are expected at each level.

An analysis of recent studies [6-19] was carried out and the most frequently used methods of ensuring Smart Parking Security were highlighted. These are: the use of blockchain technology [6, 7, 10, 14], the use of biometric security mechanisms [7], radio frequency identification (RFID) and the use of a wireless sensor network (WSN) [8, 13] based on the cloud environment. [15, 19], 5G technology [16], neural networks of general regression (GRNN) [23], fuzzy logic and undefined data [24], multivalued

logic [25]. But all these studies focus on solving one or two safety criteria of the Smart Parking System and do not provide solutions to all the above criteria in combination. Therefore, in order to improve the security of the smart parking system, it is necessary to analyze the software architecture. The purpose of this analysis is to select the parts of the software that are most vulnerable to external threats and provide a solution for ensuring the security of the smart parking system software in terms of compliance with all criteria in the complex.

Thus, the urgent task is to analyze the security requirements of the smart parking system software in order to identify parts of the software that are most vulnerable to external threats and develop methods and means to improve their security. Given the above, the purpose of this study is to develop methods to improve the security of the smart parking system, taking into account bottlenecks in the software system and parts that are most vulnerable to external threat factors.

1.3 Analysis of custom applications for smart parking

Smart parking systems are becoming more common, offering innovative solutions to manage parking and improve its efficiency. However, along with the growing popularity of such systems, new security challenges are emerging. Ensuring security in smart parking systems requires the use of appropriate software solutions. This article will analyze some software solutions that are used to ensure security in smart parking systems.

"Smart Parking Security Suite" is a comprehensive software solution specially designed to ensure security in smart parking systems. It offers a wide range of features, including video surveillance, access control system, automatic license plate recognition, and intrusion detection system. This allows the detection of illegal activities, such as the use of fake parking tags, vandalism or unauthorized access to parking lots.

"Parking Security Management System" is a software solution that provides security in smart parking systems by integrating with various devices and technologies. It includes an access control system, motion sensors, video cameras and an alert system. This allows you to track the movement of cars, detect unusual activity and alert relevant services to potential threats or security breaches.

- Video surveillance: the system provides surveillance of parking lots using video cameras, which allows you to detect unusual activity or security breaches.

- Access control system: provides authorized access to parking lots through the use of cards or other identifiers.

- Automatic license plate recognition: allows you to automatically identify the license plates of cars entering the parking lot and compare them with the database for authorization verification.

- Intrusion detection system: uses sensors and behavior analysis to detect unusual or suspicious activities such as vandalism or unauthorized access.

- Access control system: provides limited access to parking lots only to authorized users.

- Motion sensors: notify about the movement of cars in the parking lot and help track the movement of cars in real time.

- Video cameras: record video from parking lots, which allows you to track events and actions of users.

- Notification system: sends notifications to administrators or relevant services about potential threats or security breaches.

The analyzed software solutions, such as "Smart Parking Security Suite" and "Parking Security Management System", are used to ensure security in smart parking systems. They offer a wide range of functionalities, including video surveillance, access control system, automatic license plate recognition, and intrusion detection system. These solutions help detect illegal activity, provide access control and alert you to potential threats to the security of the parking system.

However, when choosing a software solution for security in a smart parking system, it is important to take into account the specific needs and requirements of a particular system, as well as ensure its integration with existing components and infrastructure. In addition, it is important to keep your software up to date and take steps to prevent new security threats.

The analysis of software solutions for security in smart parking systems is an important step in the development and improvement of safe and reliable parking systems that contribute to improving convenience and safety for users.

There are currently hundreds of parking apps available in the mobile app market. However, this section selects parking apps that provide real-time parking occupancy statistics and navigational directions to your reserved parking space. All the above applications support booking, which is only available in closed parking lots. In an enclosed car park, the smart parking software also gives navigation directions to the booked parking space. Smart parking tools consist of sensors, technologies and programs that are used to detect people in a parking lot and improve parking efficiency. Below are some sensors that help detect parking occupancy information. Sensors are a typical tool that has been investigated in detail in previous publications.

Passive infrared sensor: These sensors detect energy changes, and when the car takes up a parking space, these sensors recognize energy shift and detect employment. When a vehicle or person is standing above the sensor, the sensor detects a change in energy. It can be used to allocate emissions based on the amount of energy change. These sensors, however, are sensitive to the environment and will be inaccurate in the presence of snow or rain. Passive infrared sensors must be installed underground or on the roof. As a result, they require significant costs for the purchase and maintenance of these sensors. These sensors are suitable for covered enclosed car parks, but not for outdoor outdoor car parks. The principle of operation of the infrared sensor is shown in Figure 1.4.

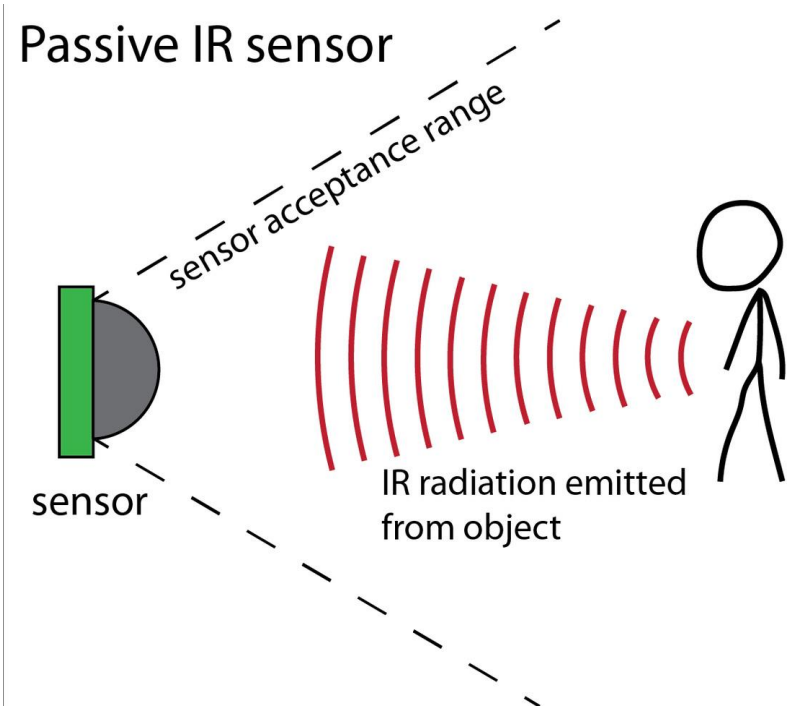


Figure 1.4 – The principle of operation of a passive infrared sensor

Active Infrared Sensor: These sensors emit infrared energy and use the amount of reflected energy to detect any object or vehicle. They are also sensitive to environmental changes such as rain or snow. As a result, they require installation in all parking spaces and require significant investment and maintenance. Installing sensors in all parking spaces will help to obtain parking employment status. These sensors are often installed overhead and are suitable for indoor enclosed car parks. Because these sensors are sensitive to environmental changes, they are not suitable for outdoor car parks. The principle of operation of the active infrared sensor is shown in Figure 1.5.

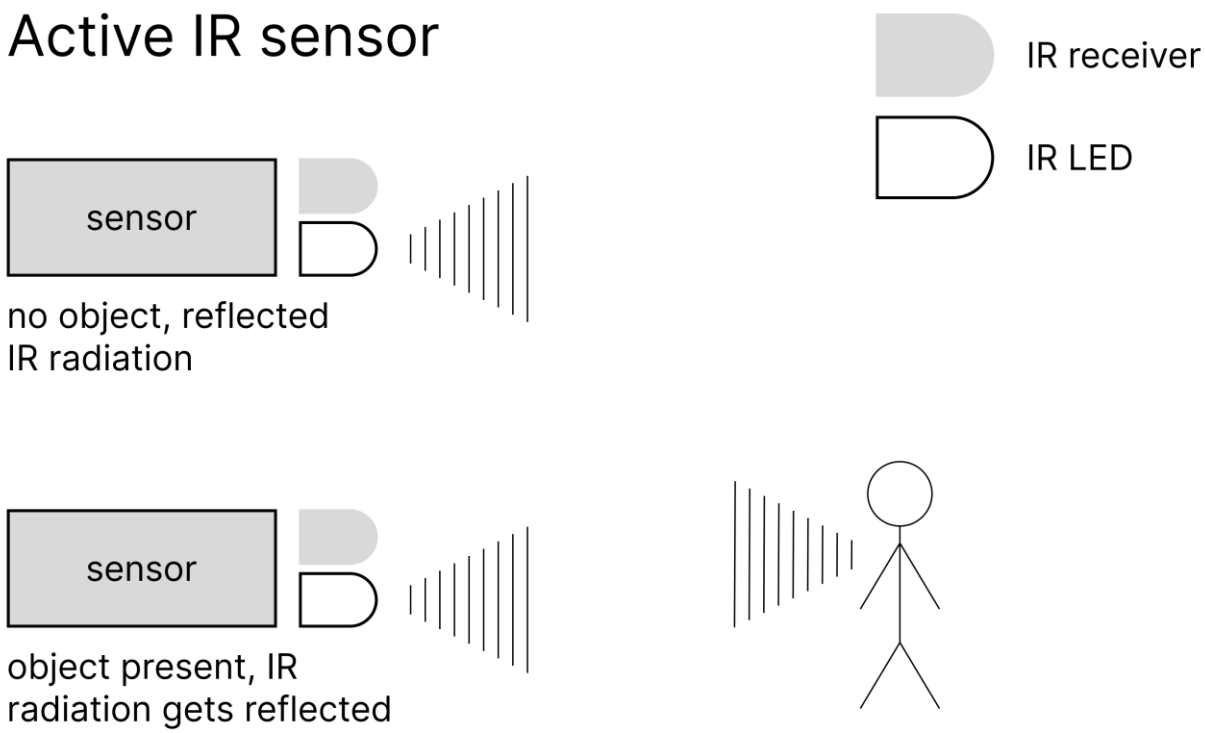


Figure 1.5 – The principle of operation of the active infrared sensor

Ultrasonic sensors: generate sound waves in the range from 25 to 50 kHz and identified objects based on reflected energy. They are often ceiling-mounted and sensitive to environmental changes such as rain and snow. As a result, they are more suitable for indoor parking lots than outdoor parking lots. It can distinguish a vehicle from a person based on the distance at which the waves are reflected. These sensors should be placed at the top of each parking space to obtain parking employment status. These sensors would be inexpensive, but installing and maintaining multiple sensors, as well as connecting them to the grid, would be expensive in the long run. Wireless ultrasonic sensors are also used to collect parking occupancy information. Wireless sensor networks, such as the ZigBee protocol or other similar networks, connect them. On the other hand, wireless sensors need regular maintenance. Another study uses ultrasonic sensors on a passing car and parking space occupancy data is collected on a regular basis. A passing car cannot get real-time parking information about occupancy.

Inductive loop detectors are installed using an underground wiring system and use electromagnetic principles to detect the presence of a vehicle. They are usually used at the entrance and exit to determine the number of cars present, which can be used to determine the availability of parking spaces. These detectors are expensive to install and maintain, but they are commonly used in covered car parks to count available parking spaces. These detectors, which are used in several commercial parking lots, offer an accurate count of cars in an enclosed parking lot. However, the occupancy status of a separate parking lot cannot be determined using inductive loop detectors.

1.4 Conclusions

So, in the first section, an overview of existing methods and ways to implement a cyber-physical system for smart parking was conducted. Among them: a method using ultramagnetic sensors, infrared sensors, blockchain technology and inductive loop detectors.

The advantages and disadvantages of existing methods and tools were also analyzed and it was concluded that the above methods are expensive and difficult to install, install and maintain, as well as difficult to scale. Therefore, for further work, a method of implementing a cyber-physical system for smart parking using outdoor surveillance cameras was chosen. This method is more economical and noticeably easier to scale and easier to install and maintain.

2 THE PROBLEM OF SMART PARKING CYBER-PHYSICAL SYSTEM SECURITY

2.1 Client-Server Architecture and Possible Security Threat Factors

Since the proposed cyber-physical system for smart parking, which is shown in Figure 2.1, consists of two parts – hardware (cameras and all physical devices necessary for functioning) and software (client and server subsystems), it is necessary to investigate possible factors, which may affect the security of this system.

Figure 1.5 – The principle of operation of the active infrared sensor

If hardware can simply be tested for reliability and performance, the software subsystem needs more in-depth research. Given that the system has both server and client parts, an analysis of factors affecting the security of both parts of this cyber-physical system was carried out. These include: incorrect security configuration, client-side injection (insecure authentication data, malware), insufficient transport layer protection (MITM attacks), insecure storage (database), device rooting/ jailbreak, reverse engineering, disclosure of confidential data (private data breaches), improper

logging and monitoring. The results of the analysis are presented in a schematic form in Figure 2.2.

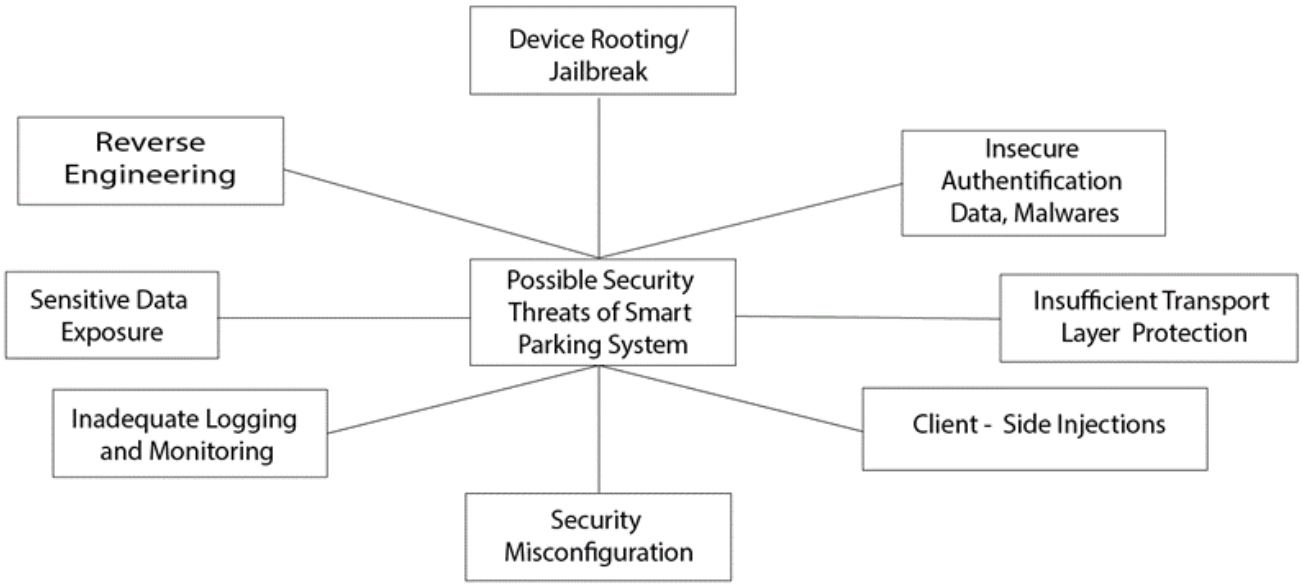


Figure 2.2 - Possible threats to the security of the cyber-physical system for smart parking

2.1.1 Client-side security risks

Since the client part should be developed in the form of a cross-platform mobile application that does not involve storing any private information, such as a personal phone number, login and password, it may seem that it is less susceptible to attacks from hackers or information leakage. However, the possibility of risks associated with the security of the system on the client side cannot be completely excluded. The server part of the software, on the contrary, is very sensitive, since it contains algorithms for recognizing car images using an artificial neural network. Unauthorized access to the database, program code or system files can lead to incorrect operation of algorithms and, as a result, to providing the client part with incorrect information about the occupancy or unoccupied parking space. That is, incorrect operation of the entire system as a whole. In addition, since communication between the client and server parts

must be implemented using an application programming interface (API), additional bottlenecks appear in the smart parking security system.

For greater user convenience and faster access to the system, it was decided to develop a client part in the form of a cross-platform mobile application. Over the past decade, the mobile app development industry has grown significantly, but cybercrime has not remained at the previous stage. All this has led to the fact that it is impossible to download a mobile application to the Google Play Store or Apple App Store without checking security indicators and making sure that the application will not be accused of information leakage or personal data fraud. But mobile app security is more than just protecting them from malware and external threats. First we need to define the basic security principles of open web applications and their main security threats in order to be able to analyze security measures and develop methods and tools to increase their level of security [3].

Improper use of smartphone functions or unforeseen failures when using security settings. This includes privacy settings, permissions, misuse of Touch ID, FaceID, Keychain, etc.

A sufficient bottleneck that can often be encountered when solving security problems of mobile applications is the lack of a secure storage system. Mobile device developers typically rely on client device storage for some personal and internal data. But if hackers gain access to the device or the device can be stolen or lost, this data can be used for malicious purposes. As a result, this leads to cybercrimes such as privacy policy violations and identity theft for malicious use.

During the development of mobile applications, data transfer takes place on a client-server model. Thus, when data is transmitted, it can be intercepted by attackers over the Internet. Attackers can also intercept data during bank transfers. Data transmission through unreliable communication channels leads to violation of privacy policies, identity theft, fraud and loss of business reputation of the company.

Attackers or bots can obtain data during authentication and infiltrate the user's account. This may result in leakage of personal information, identity theft and unauthorized access to internal user account data.

Attackers or advertising bots may have access to data that has not been encrypted or protected properly. This may result in unauthorized access to the internal data of the application, data theft, leakage of personal information of users, etc.

Attackers can intercept data during the authorization process and use it for unauthorized access to the program. As a result, this leads to leakage of personal information and loss of business reputation of the company.

Poor code quality can lead to unpredictable application crashes or numerous errors during use. In addition, it reduces application performance and can lead to excessive memory usage or slow loading of graphical elements in the user interface while running.

Attackers, gaining access to the source code, can integrate advertising or malicious scripts into it or replace parts of the code, which can lead to incorrect operation of the program , loss of some functions or replacement of certain functionality for using the program. for malicious purposes.

The risks that affect the security of mobile applications are presented in the form of a diagram in Figure 2.3.

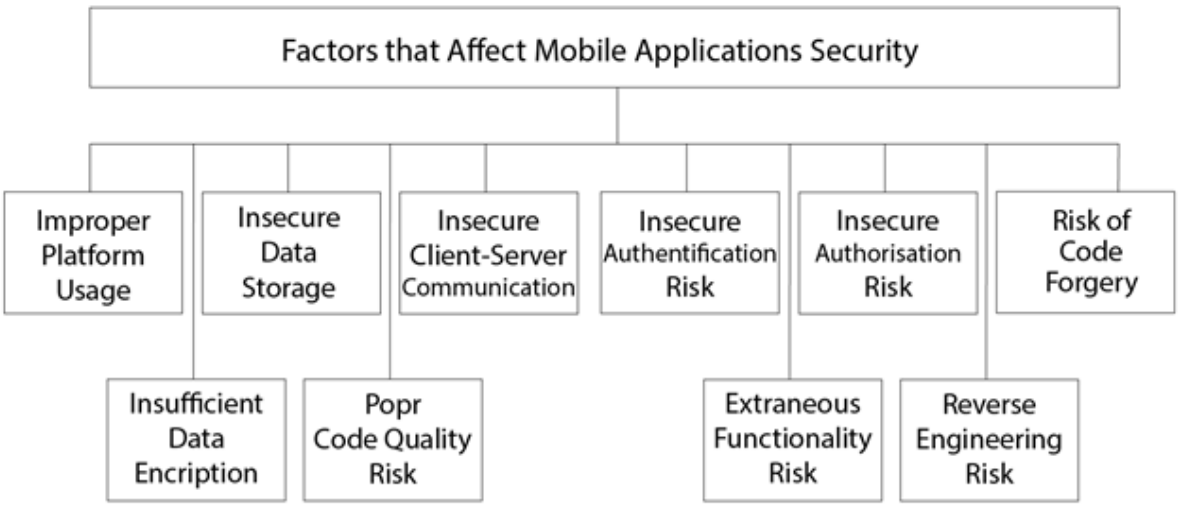


Figure 2.3 – Risks that affect the security of mobile applications

Statistics were also collected on the frequency of occurrence of each risk factor to determine the factors that occur most often, and therefore, are the most dangerous for the client part of the cyber-physical system for smart parking.

The frequency of manifestation of factors affecting the security of mobile applications is shown in the form of a bar chart in Figure 2.4.

According to the statistics presented in Figure 2.4, unsecured data storage and unsecured client-server communication are the most common causes of security risks of mobile applications.

Attackers can download a mobile app to redesign its features. That is, the same program in different versions can work completely differently.

In this case, attackers check the functions of the mobile application to find bottlenecks and introduce third-party code.

Also, withcatchers can introduce malicious changes to the mobile application, which allows them to change its functionality.

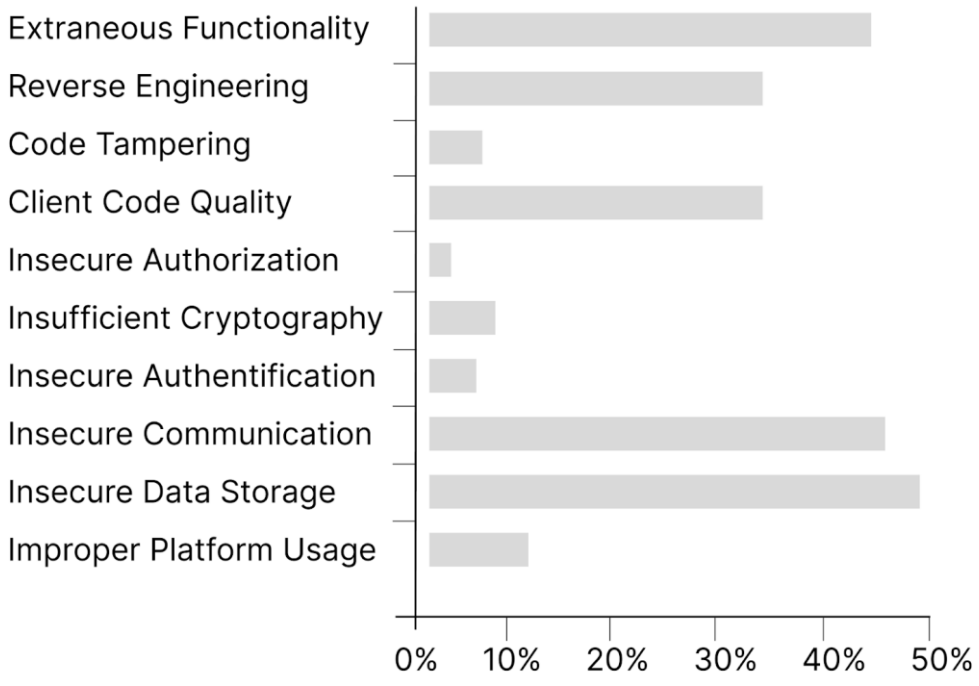


Figure 2.4 – The frequency of factors affecting the security of mobile applications

2.1.2 Server-side security risks

One of the most important factors in ensuring the security of server-side data is the importance of encryption.

Instant messaging is one of the most common ways a client communicates with a server at the client/server architecture level.

Thus, encrypted text messages are an important security measure.

Therefore, mobile app developers should pay attention to this security measure. However, information security is not the only benefit that can be gained from facilitating encrypted text messages. In fact, there are also some other ways to ensure server-side data security, which are listed in Table 2.1.

Table 2.1 – Ways to ensure data security on the server side

Way	Description
Secure data transmission	There is no denying that encrypted data ensures security during transmission. If users send files in e-mail or distribute them through a cloud-based server, they can use encryption. This helps them make sure that no unauthorized user can view this information.
Ensuring data integrity	Misuse of data can occur not only through targeted data theft, but also through manipulation. A hacker can manipulate specific data to disrupt corporate communications. However, using encrypted data can avoid this situation.
Data protection on all devices	A variety of devices is becoming a significant part of our lives. However, transferring data from one device to another is a risky business. So, encryption technology can protect and store data over all devices. Such a security measure can deter unauthorized users.

End of the table 2.1 – Ways to ensure data security on the server side

Data encryption	In most cases, it is clear that many industries need strict adherence to the protection requirements of all those whose personal information is stored by organizations. FIPS, HIPAA, or any other regulation relies entirely on encryption to protect data

2.2 API Security Risks

Application Programming interface (API) is a type of software that connects to the functionality of an application and saves developers time. Often, the API connects functionality created by other developers, or frequently used functionality, or a client-server connection. This helps developers save time and not develop features that are already generally available from scratch. When it comes to connecting several parts of a software system together, it is really useful [17]. However, there are also security risks when using the API. There are two main reasons why security issues should be considered when using the API.

1) An easy way to access the internal information of the application – through the API, you can access the stored data, including the user's personal information (login, password, etc.) for the purpose of unauthorized distribution or malicious activity.

2) An easy way for attackers to bypass security measures, even if the firewall is on. Therefore, do not neglect a well-thought-out security strategy.

There is a significant difference in security measures for traditional web and API-based web applications. This difference lies in their architecture and how they are built. Previously, web application protection only required HTTP and HTTPS port protection.

Modern applications that use several APIs and different protocols require comprehensive protection of all parts of the program, taking into account all its bottlenecks. This is especially important when the API extends its functionality,

making it difficult to manage security. In addition, when replacing APIs, pre-designed security measures need to be reviewed and reconfigured manually. The difference in the structure of API-applications makes them susceptible to external threats [4]:

APIs are usually secured with a JSON web token or API key. This allows you to protect the API and, if unusual or suspicious behavior is detected, stop access to API keys .

Protection against DDoS attacks is mainly based on the principle of rejecting requests from suspicious actors. This becomes more complicated because every traffic looks suspicious in API-based apps.

The server is responsible for communication between the application and the user behind the mobile phone screen. The main reason for server vulnerabilities is that sometimes developers don't take proper security measures and protect server connections seriously when working with the API.

According to research around the world, data breaches and breaches usually occur when insufficient logging occurs.

Unlike authentication, the authorization process in each program has its own logic, and this can often be a bottleneck for attackers. If the authorization process is not well thought out and protected, hackers can enter the system and gain access to data using an iterative method of selecting an identifier [4]. The emergence of smart technologies has revolutionized numerous aspects of our daily lives, including the way we park our vehicles. Smart parking systems, employing cyber-physical technologies, have been introduced to optimize parking space utilization and enhance convenience for drivers.

According to statistics, the main and most common factors are data breaches, accidental data disclosures, outdated APIs, denial of service, unknown or shady APIs, and account hijacking. The frequency of manifestation of the above factors that affect the security of the application programming interface (API) is shown in Figure 2.5.

That is, middleware provides additional protection of the server against suspicious or malicious requests by intercepting and checking them. And only if the request is safe, it is sent to the server for further processing.

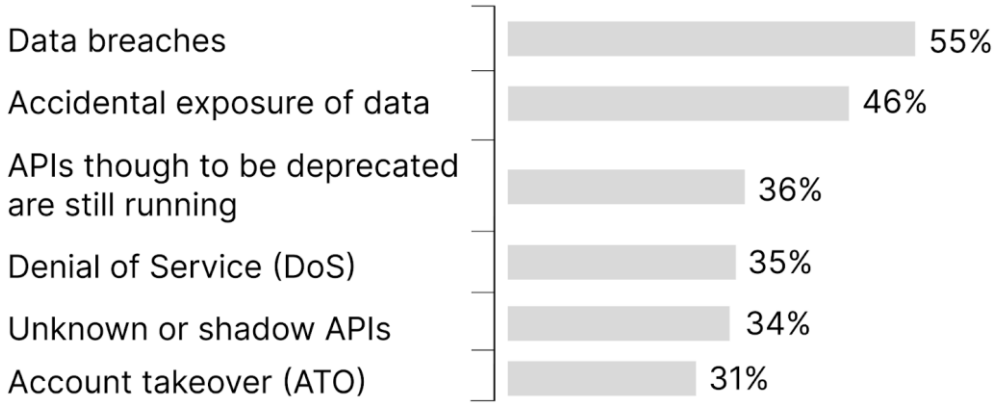


Figure 2.5 – The frequency of factors affecting the security of the application programming interface (API)

2.3 Selection of methods and environment for software implementation

Considering the factors that affect individual parts of a smart parking software system, it was decided to take into account those that most often affect the security of the system and involve its different parts from different points of view (for example, data access, client-server communication , bottlenecks when using the API) and come to a decision that will help to take them into account in combination.

The proposed solution is an intermediate software for additional validation of requests from client to server. It is an effective tool for performing operations or calculations within a request-response connection in the client-server interaction model. It should be used when you need to perform an operation or verify that the request is not trusted directly on the server for security reasons.

Therefore, the architecture of the cyberphysical system for smart parking, shown in Figure 2.1, was improved by adding middleware security to the server part of the

software. The proposed architecture of the cyber-physical system for smart parking, using middleware, is shown in Figure 2.6. With this architecture, it is much easier to determine if a request was actually sent from our user agent and to check if it is malicious and contains no suspicious code. Also, this middleware will shorten the application uptime if the request is not relevant, since its result is already known, since the Google Cloud API call does not occur instantly.

Let's consider an example of the proposed software architecture of smart parking using Security Middleware. For better understanding and clarity, consider the algorithm proposed in Figure 2.6.

According to Figure 2.6, the request is sent to the server from the mobile client application. However, the request may not always be safe. To verify this, we test the request using integrated middleware.

This will allow us to verify that the request actually came from our user agent and not from some third-party resource, and that the request is secure. Also, the use of middleware will reduce the operating time of the program if the request is unresponsive or its result is already known.

Because the Google Cloud Vision API call is not instantaneous, we may use this time to verify the security of the received request.

If the request is safe, it is sent for further processing, namely to check the parking space selected by the user for occupancy. If a request is identified as potentially harmful or extraneous, it is neutralized in the first case, and the server's response to such a request is ignored.

To develop security middleware to improve the security of the cyber-physical system for smart parking, you can use different technologies depending on the specific requirements and limitations of the developed system. The primary goal is to provide real-time information to drivers, guiding them to available parking spaces and enabling efficient parking management. Sensors deployed in parking spaces detect vehicle occupancy and relay this information to the central system, which then communicates it to drivers through mobile applications or display panels.

The list of technologies that can be used to develop middleware to ensure the security of the cyber-physical system for smart parking is shown in Table 2.2.

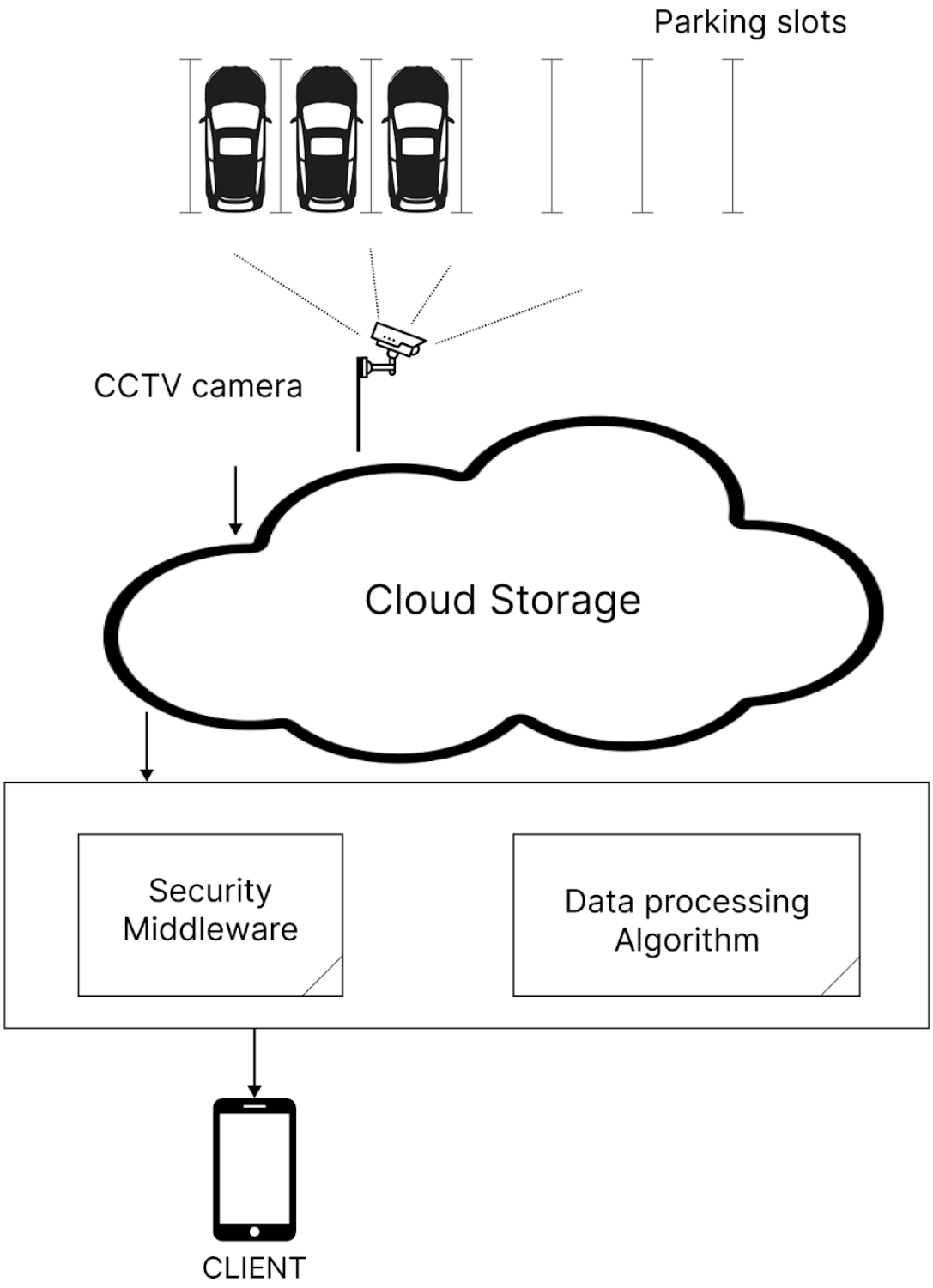


Figure 2.6 – The architecture of the cyber-physical system for smart parking, using middleware

Table 2.2 – List of technologies for the development of middleware to ensure the security of the cyber-physical system for smart parking

Type of technology	Short description
Programming language	You can use a programming language that is convenient and suitable for your use. For example, programming languages often used to develop middleware include Java, C++, Python, and Node.js.
Frameworks:	The use of frameworks can greatly facilitate the development of middleware software. For example, frameworks such as Django (Python), Spring (Java), Express.js (Node.js) or ASP.NET (C#) can be used for web programming. You can use frameworks such as Spring Boot (Java), Flask (Python) or Nest.js (Node.js) to develop microservices.
Database	Relational databases, such as MySQL or PostgreSQL, or non-relational databases such as MongoDB or Cassandra can be used to store data, depending on the needs of the system being developed.
Communication and protocols	Encryption protocols such as HTTPS, TLS, or SSH can be used to ensure communication security. Web services such as REST or GraphQL, or other protocols such as MQTT for IoT devices can be used to interact with external services or other system components.

End of the table 2.2 – List of technologies for the development of middleware to ensure the security of the cyber-physical system for smart parking

Authentication and authorization:	To ensure secure access to the system, you can use authentication and authorization mechanisms such as JWT (JSON Web Tokens), OAuth or OpenID Connect.
Monitoring and logging	To track system performance and identify possible problems, you can use monitoring tools such as Prometheus or ELK Stack (Elasticsearch, Logstash, Kibana) to help you analyze the logs and metrics of your system.
Testing	Don't forget the importance of testing your middleware. You can use automated testing frameworks such as JUnit (Java), pytest (Python) or Mocha (Node.js), or other tools for unit testing, integration testing, and load testing.

It is recommended to conduct a detailed analysis of the needs of the cyber-physical system and consult with security experts to select the best technology stack for the middleware offered.

2.4 Inconclusions

So, in this section, an analysis of potential threats that affect the security of the cyber-physical system for smart parking was conducted. Since the proposed cyber-physical system is based on client-server architecture, potential threats to the client side and server part of the software have been identified. Also, since the client part will be

connected to the server part using application software (APIs), the factors that affect API security were considered.

Also during this section, a technology stack was identified that can be used to develop middleware to improve the security of the cyber-physical system "Smart Parking".

3 IMPLEMENTATION OF SOFTWARE AND TECHNICAL TOOL FOR INCREASING THE SECURITY OF SMART PARKING CYBER-PHYSICAL SYSTEM

3.1 The proposed method of increasing the Security of Smart Parking Cyber-Physical System

Since in the previous section the technology stack was considered, which makes it possible to develop middleware to improve the security of the cyber-physical system "Smart Parking", the ASP.NET Core framework was chosen among the proposed technologies. One of the most important features of ASP.NET Core is the Middleware structure. This is a very effective feature for performing individual operations inside the request-response model and managing request-response traffic. We can perform many different tasks, such as checking the validity of an incoming request, creating responses from the cache in ASP.NET Core using structures that come together.

A structure that allows you to execute methods added as an addition to a class derived from the IApplicationBuilder interface based on middleware. Intrusion Detection Systems: Smart parking systems should incorporate intrusion detection systems that can identify and respond to suspicious activities.

We use it when we want to perform various operations and give a different direction to the process until the request is answered from the client to the web application.

Middleware is a terminology structure that describes middleware. This is a structure that has the same mechanism of work everywhere. This includes advising them to install security updates promptly, avoid connecting to unsecured networks, and report any suspicious activities or system malfunctions.

The middleware starts slowly. When middleware is started, it starts other middleware before it expires. The scheme of work of components of middleware to improve the security of the cyber-physical system "Smart parking" is shown in Figure 3.1.

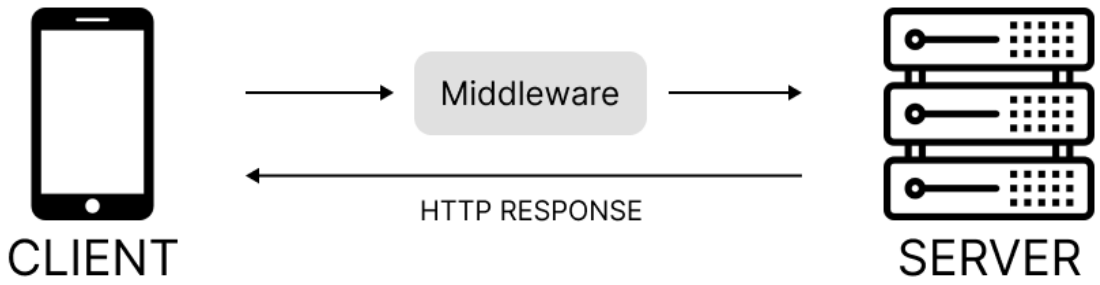


Figure 3.1 – Scheme of operation of the components of the cyber-physical system "Smart parking" taking into account the middleware

The principle of operation of the middleware to improve the security of the cyber-physical system "Smart Parking" with the client part of the cyber-physical system, which is offered in the form of a cross-platform mobile application, is to receive requests from the client to the middleware and provide an answer with already verified and secure data. If the data for some reason is not verified by middleware, no response is provided to such a request, and the data of the potentially malicious request itself is destroyed by the middleware. The principle of operation of middleware to improve the security of the cyber-physical system "Smart parking" is shown in Figure 3.2.

In the diagram shown in Figure 3.2, a request is received and one middleware is running, and the requested operations are performed at this point, and then the next command runs the next middleware. It continues to run in this way until the third middleware, then because there is no middleware to run, it returns to the previous middleware, completes it and returns to the previous one, and when the last one is completed , the answer is returned. Thus, the spiral is completed.

Asp.Net Core has a kernel that structurally supports structuring middleware. All functions in the Configure method in the startup file actually act as middleware. In structuring Asp.Net Core, middleware starts with the name "Use" and is called in the settings.

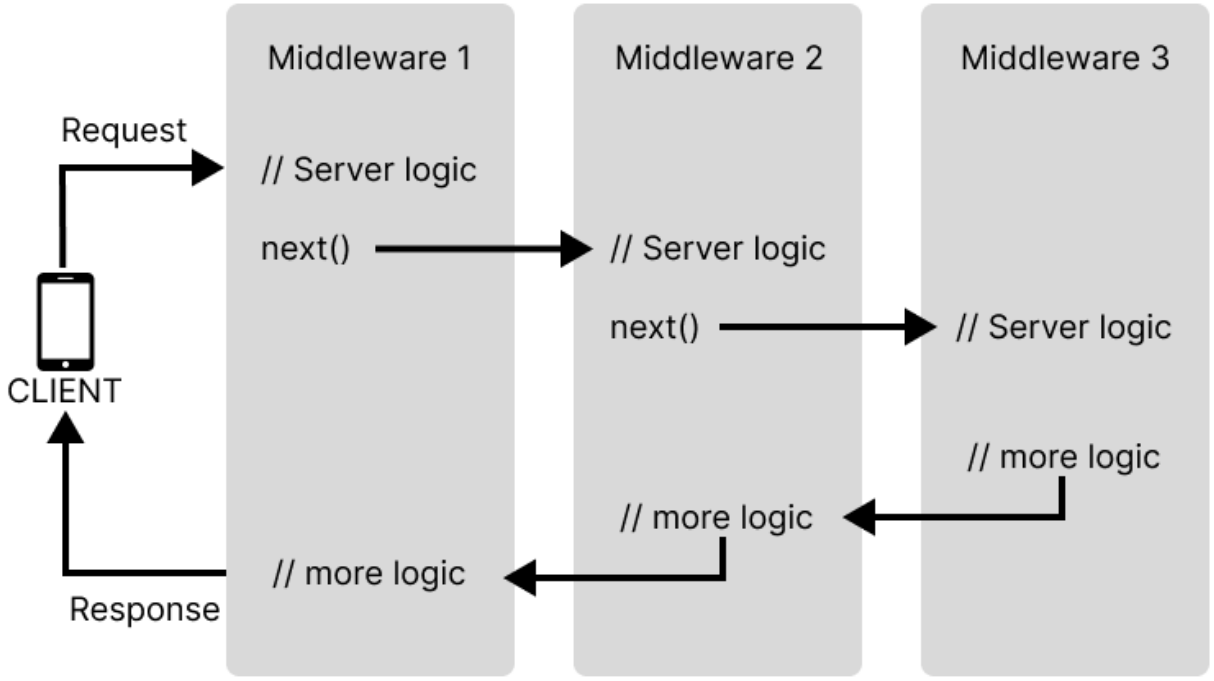


Figure 3.2 – The principle of operation of middleware to improve the security of the cyber-physical system "Smart parking"

The order of triggers is important in Middlewares. Therefore, the middleware should be sorted paying attention to the working priority. For example, if you need to authenticate and authorize, you must call middleware for authentication first , followed by middleware for authorization. Ignoring the order here will lead to logical errors.

Asp.Net Core has middleware installed in the kernel: Run, Use, Map, MapWhen. Table 3.1 describes the classes and modules of middleware to enhance the security of the cyber-physical system for smart parking on the ASP.NET Core platform.

Table 3.1 – Description of classes and modules of intermediate software to improve the security of the cyber-physical system for smart parking.

Class name	Description and purpose
SecurityMiddleware	The main class of middleware that performs security checks before processing a request. This class implements the

	interface

Continuation of Table 3.1 – Description of classes and modules of intermediate software to improve the security of the cyber-physical system for smart parking.

	'IMiddleware' and contains the method 'InvokeAsync', in which the security logic is executed and the next processing step is called.
AuthenticationService	The class responsible for user authentication. It contains methods for verifying user identity, such as login and password, and returns a 'ClaimsPrincipal' object with an authenticated user.
AuthorizationService	The class responsible for authorizing users. It contains methods for verifying user access rights, such as based on roles, permissions, or other criteria. It returns a logical value indicating whether the user has the necessary access rights.
UserContext	A class that represents the user's context. It contains information about the authenticated user such as ID, name, roles, etc.
Controllers	A module that contains controllers that handle user requests. Controllers can use 'AuthenticationService' and 'AuthorizationService' to verify security and resource access.

End of the table 3.1 – Description of classes and modules of middleware to improve the security of the cyber-physical system for smart parking.

Models	A module that contains data models required for authentication and authorization, such as a user model or access rights.
Extensions	A module that contains extensions for registering services and setting up middleware in ASP.NET Core. For example, you can create an extension method that adds 'SecurityMiddleware' to a query handling pipeline.

Methods of middleware classes to improve the security of the cyber-physical system "Smart parking" are presented in Table 3.2.

Table 3.2 – Methods of middleware classes to improve the security of a cyber-physical system for smart parking

Method name	Description
Startup module	Runs middleware. As a result, pupline does not continue and gives a direct output. This effect is called short circuit. It can be use in accordance with the operation that will be performed.

Use	The usage method calls the following middleware in the process after its activation and has a structure that can go back and continue working after the normal middleware function has completed.

End of the table 3.2 – Methods of middleware classes to improve the security of a cyber-physical system for smart parking

Map	Sometimes, we may need to filter the middleware according to the path that sends the request. For this purpose, we can provide control in the "Use" or "Execute" functions, or perform more professional operations using the "Map" method.
MapWhen	With the Map method, filtering is performed only according to the path on which the request was made, filtering is performed according to any function of the incoming request using the MapWhen method.

To unify and create a common mechanism for intercepting errors using middleware supplied with .NET Core and registering received errors, a project was created as WepApi and a new class LoggingMiddleware in it. It consists of a constructor and an Invoke method that accepts a simple RequestDelegate parameter. We will catch the exception by sending a request to the try catch block and write the tracked error to a txt file.

3.2 Structural diagram and algorithm of operation of Software and Technical Tool for increasing the Security of Smart Parking Cyber-Physical System

When developing the architecture of the cyber-physical system for smart parking, it was decided to divide the entire architecture of the system into three levels - the external level, the intermediate level and the internal level.

On the outer level there are parking markings and outdoor surveillance cameras from which data will be collected.

At the intermediate level there is maintenance equipment, namely a server, a cloud database and an Internet connection, through which communication between the components of the system takes place. The internal level includes software that supports the operation of the system and client-server interaction.

Among the key software, intermediate software to improve the security of this cyber-physical system and data processing algorithm were highlighted.

The software architecture of the cyber-physical system for smart parking using middleware to improve security is shown in Figure 3.3.

The software architecture for the cyberphysical smart parking system using middleware to enhance security can be organized using layer or microservice architecture. General description of the architecture of the cyber-physical system for smart parking, which is shown in Figure 3.3:

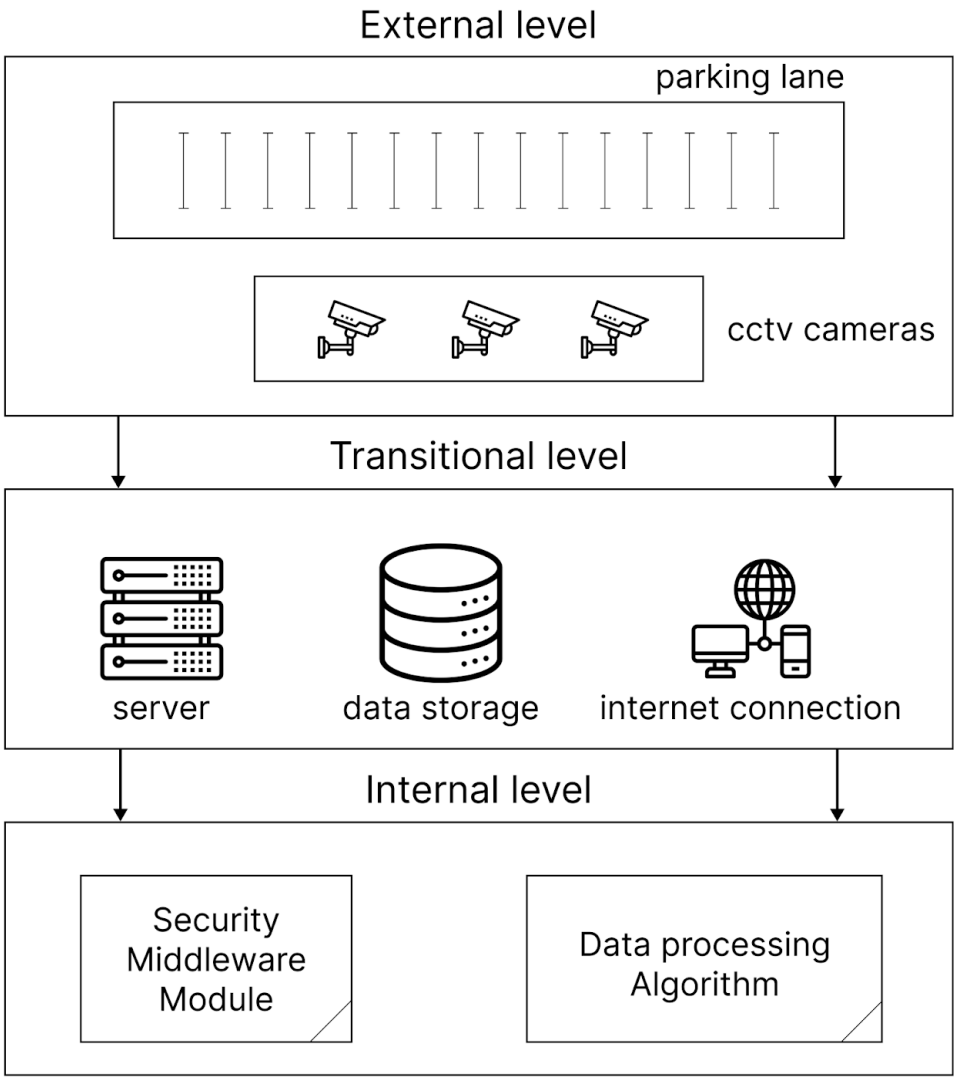


Figure 3.3 – Architecture of the cyber-physical system for smart parking using middleware

1. The client interface includes a web application or mobile application that users can manage smart parking. This interface provides user interaction, sends requests to the server, and receives responses.

2. Server side: includes middleware that ensures system security and communication between different components. Middleware security is the heart of the system, which includes various security components such as authentication, authorization, encryption, access control, and logging.

It performs security checks and protects the system from potential threats.

The control module is responsible for managing and coordinating all components of the smart parking system.

It accepts requests from the client interface and performs the necessary actions, including processing parking operations, monitoring the status of the system and ensuring security. Parking components include sensors, barriers, cameras, etc. that provide parking status data collection and access control.

They interact with the control module to transfer data and perform appropriate actions.

The database stores information about users, parking lots, transactions and other important data. Used to ensure permanent storage and access to data.

3. Security and protection. This component includes all the security measures necessary to protect the system. It uses middleware security software for access control, authentication, authorization, encryption, and logging.

4. Monitoring and analysis. This component provides monitoring and analysis of the system state, detection of anomalies and security events. It can include logging systems, network monitoring, intrusion detection, etc.

It is especially important to consider security principles such as data protection, authentication, authorization and encryption, and implement them in accordance with the architecture of the cyber-physical system for smart parking.

Figure 3.4 proposes an algorithm for checking the security of a request from client to server using application software.

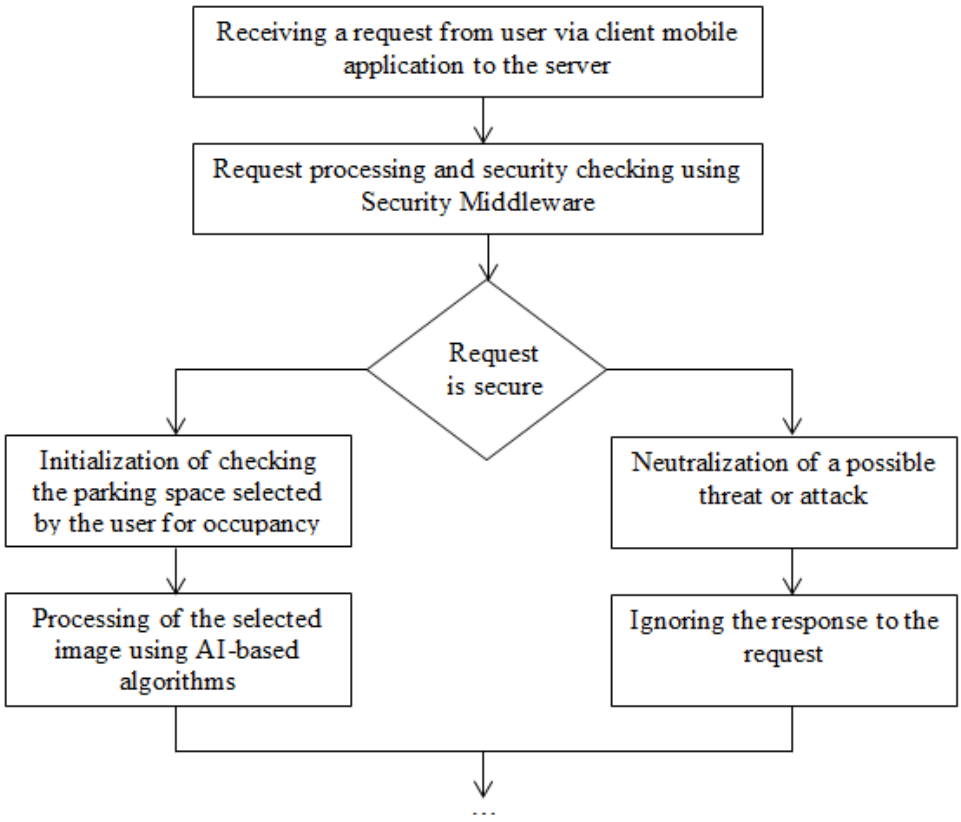


Figure 3.4 – Algorithm for checking the security of the request using middleware

The algorithm for checking the security of a request from client to server using middleware in a cyber-physical system for smart parking may include the following steps:

1. Acceptance of the request. The middleware receives a request from the client to the server.

2. Authentication. The request is checked for correct authentication data, such as a token or user ID. This may include verifying your login and password, or using access tokens such as JWT (JSON Web Token).

3. Authorization. After authentication, the request is checked for access to the resource or the execution of a specific action. Access rights, user roles, or other criteria are used to determine whether the client has the necessary rights to execute the request.

4. Encryption. If sensitive data such as passwords or personal information is transmitted, it can be encrypted to ensure privacy while in transit over the network.

5. Access control. Access permissions to a specific resource or operation are checked. This may include checking the permissions of the database or other internal resources.

6. Logging. Information about the request and its result can be recorded in a log file for further analysis and monitoring of system security.

7. Sending the request to the next level of processing. If the request successfully passes all security checks, it is passed to the next processing level, where the required business logic is executed.

It is also important to use robust cryptographic algorithms and perform regular updates of middleware to ensure system security.

When the user enters the mobile application to search for free parking spaces in the "smart parking" with a video surveillance system, the server receives an HTTP request. Next, with the help of the Security Middleware software component, the handler on the server verifies the request for security. Suspicious or potentially dangerous requests are blocked at this level to further protect the system against malicious code, SQL injections, and possible data leaks or source code changes on the server.

If the request is safe, it is sent for further processing by the server. The parking lot selected by the user is checked for free parking spaces, and their status is visualized using colors. For this, image recognition algorithms based on artificial intelligence from the Google Cloud Vision library are used. This library interacts with the processor on the server through an API.

If the car is recognized, the parking space is considered occupied and is indicated by a red rectangle on the user interface of the mobile application. If the car is not recognized, the parking space is considered free and is indicated by a green rectangle on the user interface.

In addition, by clicking on the marked parking space, the user can view the video from the external surveillance camera of the selected parking lot. This allows you to see

in advance how it is more convenient to enter the parking space and monitor the car in real time to ensure additional safety of the vehicle in the parking lot.

The diagram shows the interaction process between the server and client parts of a cyber-physical smart parking system that uses image recognition technology using an artificial neural network. The top left corner of the diagram shows the Google Cloud Vision API image processing server, which was chosen after the experiments in Section 2. The bottom left corner shows the client device, which can be a smartphone with an operating system no older than Android 7. The center shows a physical server with Linux operating system and Apache Tomcat Server web server. On the right side, the database server with the MySQL DBMS is depicted.

Continuing the process, after displaying information about the status of parking spaces on the user interface, the user can click on the marked parking space to view the video from the outdoor surveillance camera. This gives the user the ability to see the best way to pull into a parking space and real-time monitoring of their vehicle for added safety.

This process is implemented thanks to the interaction of the server and client parts of the cyber-physical system. The image processing server uses the Google Cloud Vision API, which provides recognition of cars and determination of their status (occupied or free). A client device, such as a smartphone, interacts with the server through a mobile application.

The physical server based on the Linux operating system and the Apache Tomcat web server are responsible for processing requests, transferring data between the server and the client device, as well as managing the interaction with the MySQL database, which stores the necessary information about parking spaces and their status.

All these components work together to provide the user with a convenient and safe procedure for finding and using free parking spaces on "smart parking". This allows efficient use of available resources and improves the user experience when parking their vehicle.

The smart parking system, which uses images from outdoor surveillance cameras, is based on a pattern recognition algorithm based on an artificial neural network. When developing a system, the question arises: to use a ready-made algorithm in the form of a library or to develop your own algorithm for learning an artificial neural network. Due to the significant expenditure of time, hardware and software resources and the need for specialized equipment, a decision was made to use a ready-made software library.

To develop the server part of the software, based on the image recognition algorithm using artificial intelligence, the following work plan was drawn up:

1. Analysis of known technologies and artificial neural networks for image recognition.

2. Testing existing neural networks on available datasets of car parking images.

3. Selection of the most effective technologies based on the results of the analysis.

4. Collection of own dataset consisting of images taken from outdoor surveillance cameras of parking lots of Khmelnytskyi National University.

5. Testing of selected neural networks on our own datasets containing real images of cars from the parking lot of the Khmelnytskyi National University.

6. Selection of the most effective technology based on the test results for further work during the development of the server part.

Among the well-known and widely used technologies for image recognition are artificial intelligence-based libraries such as OpenCV [27], Google Cloud Vision [26], and the Detectron2 library [32], which is written in the Python programming language. For the initial work, it was decided to use available datasets of parking images [28-31] to test the effectiveness of each of the ready-made image recognition algorithms using artificial intelligence.

OpenCV, Google Cloud Vision and Detectron2 libraries were used for the experiment. Before starting the own learning process, it was decided to test the effectiveness of each of these algorithms on already prepared datasets of parking images that were openly available [28-31].

Various performance metrics were analyzed, such as recognition accuracy, speed, and robustness to various lighting and interference conditions. From the test results, the most effective algorithms were chosen for further work.

After choosing the most effective technologies, the collection of our own dataset was carried out, which consisted of images taken from the outdoor surveillance cameras of the parking lots of the Khmelnytskyi National University. This dataset was used to further test selected neural networks and evaluate their performance in real-world conditions.

From the results of testing on our own datasets, the most effective algorithm was chosen, which was then used in the development of the server part of the smart parking system software.

So, with the help of analysis of known technologies and artificial neural networks, testing on available datasets, own dataset and selection of the most effective technology, the server part of the smart parking system software was developed, which is based on the image recognition algorithm using artificial intelligence.

Continuing the development of the back-end software for the smart parking system, the most effective artificial intelligence-based image recognition technology was selected. This technology has been integrated into the server's software code to send requests to a neural network to recognize cars in images.

Upon receiving a request from the user, the processor program on the server performs the following steps:

1. Receives a request for recognition of free parking spaces using external surveillance cameras.

2. Runs an image recognition algorithm based on artificial intelligence, which processes images from the camera and determines the presence or absence of cars in parking spaces.

3. If the car is recognized, the application marks the parking space as occupied and marks it with a red rectangle on the user interface in the client application.

4. If the car is not recognized, the program marks the parking space as free and marks it with a green rectangle on the user interface in the client application.

In addition, the user has the opportunity to view video from the external surveillance cameras of the selected parking lot by clicking on the marked parking space on the interface. It allows the user to pre-assess the availability of a parking space and monitor their vehicle in real time to ensure the safety of the vehicle.

In general, the developed back-end software uses an AI-based image recognition algorithm to effectively detect occupied and vacant parking spaces. This allows for faster and more accurate identification of available parking spaces and facilitates the process of finding a free parking space for users of the smart parking system.

No customer information is stored on the server. Each request contains all necessary information for processing, and session data is stored entirely on the client side.

Each response must indicate whether it can be cached and for how long. If the response can be cached, the client can use the same data without contacting the server in subsequent requests. This improves performance and availability.

REST uses a multi-tier system where components can only see the components that are in the nearest layers and with which they interact.

Code on demand is an optional feature that allows the client to download and execute code.

REST is different from other protocols such as SOAP and RPC, which are messaging protocols. REST is an architectural style that sets requirements for building RESTful web services.

Compared to other API types, REST is a more flexible architecture that offers simple guidelines, allowing developers to implement requirements in their own format. REST is also highly performant, which is especially important for mobile devices.

The web component requires a server-side programming language such as PHP. For PHP, it is recommended to use the Laravel framework, which is easy to use and

reliable. Laravel offers a rich set of functionality and helps speed up web application development.

Composer is a tool that allows you to manage dependencies and libraries in a project. It makes it easy to install third-party libraries by specifying them in the composer.json file. Laravel uses the Artisan command line, which provides a set of commands for building web applications.

In addition to the Laravel framework, you can also use other popular server-side programming languages to create RESTful web services, such as Node.js (with the Express or Koa frameworks), Python (with the Flask or Django frameworks), or Ruby (with the Ruby on Rails framework) . Each of these frameworks has its own features and specific syntax, so the choice will depend on your preferences and knowledge.

RESTful web service development often uses different data formats to exchange information, including JSON (JavaScript Object Notation) and XML (eXtensible Markup Language). JSON is the more popular format because of its simplicity, ease of reading, and support for most programming languages.

In the process of developing RESTful web services, it is important to follow REST best practices and principles, such as using the correct HTTP methods (GET, POST, PUT, DELETE) to interact with resources, the correct use of HTTP status codes, creating clear resource names in URLs, use of authentication and authorization to protect access to resources, etc.

Additionally, various tools such as Postman, Insomnia or curl can be used to test RESTful web services. They allow you to send requests to a web service and inspect the responses to verify correct operation.

This is general information about building RESTful web services. If you have more specific questions or need further clarification, please let me know and I will be happy to provide you with more details.

A resource is an important abstraction of information. Any type of information that can be used can be a resource, such as a document, image, or service. The state of

the resource at a certain point in time is called the representation of the resource, which consists of the following components:

- Data: the actual information that represents the resource.

- Metadata: information describing the resource data.

- Hyperlinks: Links that help customers move to the next state.

This information can be delivered to the client in various formats such as JavaScript (JSON), HTML, XML, Python, PHP or plain text. JSON is the most common format because it is readable by both machines and humans and is independent of a particular programming language.

To access the resource, the client sends a request that contains the following components:

- HTTP method: specifies what to do with the resource (eg GET, POST, PUT, DELETE).

- Endpoint: A URI that indicates where and how to find the resource.

- Headers: contain additional information related to the client and server, such as authentication data, response format, etc.

- Body: Contains additional information to be passed to the server, such as data to be added or changed.

REST is not tied to a specific technology or platform and is not limited to a programming language. It uses six architectural constraints that must be met for an interface to be called RESTful.

Here are six architectural constraints of REST:

1. Client-server: The system must be divided into clients and servers. There can be many clients and servers, and they can develop independently of each other. This allows for easier scaling, shared responsibility and improved independence.

2. Stateless: The client must contain all the necessary information to understand and process the request. The server does not store any information about the state of the client between requests. Each request is considered independent of previous requests.

3. Caching: Clients or intermediate servers can cache server responses to improve performance and reduce server load. Server responses must include appropriate headers to allow clients and proxies to manage caching.

4. Uniform interface: All resources must be accessible through a single and unified interface. This makes it easier for customers to understand and use the system. The basic principles of the uniform interface include resource identifiers (URIs), manipulation of resources using views, self-computing hypermedia links, and the use of standard HTTP methods such as GET, POST, PUT, and DELETE.

5. Layered system: The system can be divided into layers, where each layer can be an independent component with its own functionality. Clients should not be aware of layers other than the lowest they interact with directly. This makes it easier to modify and replace individual layers without affecting others.

6. Code on demand: Optional ability to send code from the server to the client for execution on the client device. For example, the server can send JavaScript code that the client browser can execute. This allows you to expand the functionality of the client without the need to install additional software.

These architectural constraints help create a system that is simple, scalable, and provides component independence. REST can be used to develop web applications, web services, and APIs that allow different systems to interact with each other.

The server part of the cyber-physical smart parking system software has the following functionality:

1. Recognition of images from an external surveillance camera: The server part must be able to analyze the images coming from the camera and recognize the presence of cars or their absence in parking spaces.

2. Safety check of HTTP requests: The server part must include middleware that ensures safety check of all incoming requests sent from the client program. This protects the system from unwanted or malicious requests.

3. Real-time monitoring: The server part must have the ability to connect to an external surveillance camera and receive real-time images. This allows the system to

monitor cars and provide up-to-date information on the availability of free or occupied parking spaces.

4. Providing a conclusion about the status of parking spaces: Based on recognized images and data analysis, the server part has the ability to determine whether a parking space is free or occupied. The recognition result should be provided in the form of a response indicating the status of each parking space.

So, the main functional requirements for the server part of the software for the smart parking system are as follows:

1. Recognition of images from an external surveillance camera.

2. Checking the safety of requests from the client program.

3. Real-time mode for monitoring cars.

4. Providing a conclusion on the state of parking spaces (vacant or occupied).

Additional functional requirements for the server part of the software for the smart parking system may include:

5. Database management: The server part must have the ability to store and manage data about parking spaces, including their status (free or occupied) and other relevant information. The database must be available to store and update data.

6. User management: The server part may be able to manage users of the system, in particular, registration, authentication and authorization of users. This allows you to set access restrictions to system functions for different types of users.

7. Event logging: The server part can implement the event logging functionality, which records all important events and actions taking place in the system. It helps track system usage history, identify issues, and analyze events to improve performance and security.

8. Scalability and reliability: The server part must be able to work in scalable conditions, providing efficient processing of many simultaneous requests and ensuring high availability of the system. This can be achieved through scale-out, server redundancy, and other high-reliability strategies.

9. Integration with the Google Cloud Vision library: Since the Google Cloud Vision library is already used for image recognition, the backend must provide interaction with this library via API. This includes transmitting images from the camera, receiving recognition results and processing them to determine the status of parking spaces.

10. Ensuring security: The server part must have protection mechanisms against unauthorized access, hacking and attacks on the system. This includes the application of encryption, user authentication, access control and security monitoring.

These functional requirements will help to ensure efficient and secure operation of the server part of the cyber-physical smart parking system software.

Module driver: The driver module controls the communication with the hardware devices. The sensor driver handles the belts, the signaling driver manages the proximity transceiver of the vehicle, and the IFD driver is responsible for car identification and is part of the driver module.

Communication module: It receives and sends messages between the sender and receiver. This module emulates the communication process and performs error control, such as checksum verification and error correction, for the vehicle-to-infrastructure (V2I) communication. Since the communication response time is limited, the goal is to improve communication speed and message accuracy. This module transmits messages between two fixed transceivers for infrastructure-to-infrastructure (I2I) communication, such as the transceiver at the parking lot and the transceiver in the booth.

Functional module: This is the core function of the parking system, which includes monitoring, registration, reservation, and advertisement management. The functional module can communicate with hardware devices and transmit/receive data without being aware of the intricacies of lower levels, thanks to its subscription to the driver module and communication module.

Application module: It controls the entire parking system. The main functions of the application include account administration (cash and credit/debit management), operation management, fault tolerance, and service management.

Please note that the provided translations are approximate and may require further refinement based on the specific context and terminology used in your application.

Magnetic and infrared sensors are not used so often because they are not practical. The magnetic sensor is easily damaged, although despite this, it could be perfectly suited for the implementation of smart parking technology. The infrared sensor is less practical and is not used as often. It is more practical for home use.

Advantages and disadvantages are present in each of the above technologies, because they work under different conditions and according to different principles. To build a dynamic smart parking system, parking spaces must be determined automatically. This task can be accomplished using an object detection algorithm.

An infrared sensor is not practical for a project like this, as it is often affected by external factors. The purpose of data transfer in the system is to transfer data to a server where parking information can be registered. The transmitted data is intended to provide information about the parking status, i.e. how many parking spaces are available and how many of these parking spaces are unoccupied.

The operation of this sensor can be affected by the external environment, weather conditions, or people who just pass by and accidentally fall into the infrared sensor's operating zone.

The camera is quite a practical technology that is used constantly and regularly. Thanks to this technology, not only films are shot or moments from one's own life, which as a result are shared on social networks. Also, thanks to the cameras, video surveillance is carried out, which in turn transmits the image to the device.

This technology is quite often used. But it also has its drawbacks. Due to an unstable connection, the image and quality may fail. Also, the disadvantages include the high price of such equipment.

Software requirements are a set of requirements for the properties, quality and functions of software that will be developed or is under development. Requirements are determined in the process of requirements analysis and recorded in the requirements

specification, precedent diagrams, and other artifacts of the requirements analysis and development process [18].

Functional requirements (Functional Requirements) are software requirements that describe the internal operation of the system, its behavior: data calculation, data manipulation, data processing and other specific functions that the system must perform [19], that is, in other words, functional requirements determine what what a software product should do and what its individual parts are responsible for. These requirements are described in the Software Requirements Specification document.

Unlike non-functional requirements, which define what a system should be, functional requirements define what the system should do. Functional requirements for software are determined at the first stage of the software development process - at the stage of requirements analysis[19].

Since the purpose of this work is to design the server part for the cyber-physical system "Smart Parking", we will consider the functionality that relies on the server part.

Non-functional requirements (Non-Functional Requirements) are software requirements that set criteria for evaluating the quality of its work. Unlike functional requirements, which define what the system should do, non-functional requirements define what the system should be. Non-functional requirements for software are determined at the early stages of the software life cycle, namely at the stage of analysis of requirements for the software system being designed [20].

Non-functional requirements can be divided into two categories:

– Aimed at improving the system (security, reliability, speed, ease of use);

– Aimed at improving (scalability, reproducibility) system properties.

Since in this work we do not consider the client part of the system, we do not take into account the requirements for the user interface. However, when designing the server part, you should not reject the requirement for the convenience of the software interface for interaction with the client part and the database and communication through the HTTP request.

Hardware and software requirements, that is, a description of the hardware and software platforms necessary for the operation (and support) of the system. In our case, these are the technical characteristics of outdoor surveillance cameras (high resolution, the ability to record video in the dark, protection against adverse weather conditions), constant access to the Internet and uninterrupted power for continuous access to the server, the server's operating system - Ubuntu, the presence of a database .

Security and privacy requirements of the server side also play an important role, since unauthorized access to the server can lead to algorithm errors and incorrect program results, namely incorrect image recognition and incorrect conclusions about the occupancy or non-occupancy of a parking space, data leakage or malfunctions system as a whole. Also, when designing the server part, it is important to comply with the requirements for reliability, that is, for the system to work without failures 24/7. And in the event of a possible failure, the operation of the system and data could be easily restored, that is, the requirement of recoverability. The requirement for the performance and speed of operation of the server system is also quite important, because the speed of obtaining results by the user and the reputation of the system itself depend on this.

Since the smart parking system based on outdoor surveillance cameras can be used both for public use and for commercial organizations, its environmental friendliness and safety of interaction with the environment are also important non-functional requirements. Since for the operation of the server part we only need external surveillance cameras, which are often already equipped in parking lots near large shopping centers, there is no need to install additional equipment, such as magnetic or infrared sensors. Therefore, there is no need to carry out additional construction and installation work, which can damage the paving slabs or the road or parking lot surface. Standard parking markings are also required for the AI-based parking space recognition algorithm to work. Parking lots of commercial institutions and institutions are often already equipped with such markings, as well as external surveillance cameras. This is the advantage in terms of environmental friendliness and environmental friendliness of the smart parking method based on external surveillance cameras, as well as its

economic effect, compared to other known methods. It is also worth noting the safety advantage of this type of smart parking. Thanks to the external surveillance cameras and the functional requirement for the ability to monitor the car in real time (connection to the external surveillance camera), the user through the client part of the system - the mobile application will be able to monitor the car, for example, while shopping in a shopping center or in a movie theater or being in university

The following stack of technologies was chosen to implement the server subsystem of the cyber-physical smart parking system in order to test the operation of image recognition algorithms from an external surveillance camera:

1. Python programming language and the Flask framework for working with the OpenCV library.

2. OpenCV computer vision library for camera image recognition.

3. PHP programming language and Laravel framework for working with REST API.

PHP Storm and PyCharm IDE environments will be used to develop the server part.

OpenCV was started at Intel in 1999 by Gary Bradsky, and the first release was released in 2000. In 2005, OpenCV was used on the Stanley car that won the 2005 DARPA Grand Challenge. Later, its active development continued with the support of Willow Garage under the leadership of Gary Bradsky and Vadym Pisarevskyi. OpenCV now supports many algorithms related to computer vision and machine learning and is expanding every day.

The OpenCV library supports a wide range of programming languages such as C++, Python, Java, etc. and is available on various platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also being actively developed. However, OpenCV-Python was chosen for this work as one of the simplest and easiest to use options. It is a Python API for OpenCV that combines the best qualities of the OpenCV C++ API and the Python language.

OpenCV-Python is a Python bindings library designed to solve computer vision problems. Python is a general-purpose programming language started by Guido van Rossum that quickly became very popular, mainly because of its simplicity and code readability. This allows the programmer to express ideas in fewer lines of code without reducing readability.

Compared to languages like C/C++, Python is slower. However, Python can be easily extended with C/C++, allowing us to write computationally intensive code in C/C++ and create Python shells that can be used as Python modules. This gives us two advantages: firstly, the code is as fast as the original C/C++ code (because it's actual C++ code running in the background), and secondly, it's easier to code in Python than in C/ C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation. OpenCV-Python uses Numpy, which is an optimized library for numerical operations with a MATLAB-style syntax. All OpenCV array structures are converted to and from Numpy arrays. It also facilitates integration with other libraries that use Numpy, such as SciPy and Matplotlib [38].

Also, the Flask framework will be used to work with the web part, which is a Python web framework created with a small core and an easy-to-extend philosophy. It's easy for beginners to get started with Flask because there's a little boilerplate code to get a simple app up and running.

API is an abbreviation for Application Programming Interface, which means the program interface of the program. An API represents a set of rules and functions that allow two different programs to interact with each other. Such interfaces facilitate the integration of applications, enabling developers to create powerful digital solutions. An API is an intermediary between applications, sending requests and responses [39]. To send and receive the same requests (functions) and depending on the purpose of the API, different protocols and standards are followed. The following types of protocols and architectures are distinguished:

1. XML-RPC — allows the exchange of functions between two or more networks. XML-RPC uses XML to describe requests and responses, and uses HTTP protocols to transfer information from the client to the server.

2. JSON-RPC is a lightweight XML-like RPC. The protocol is encoded in JSON, which allows you to receive calls to the server with the possibility of asynchronous responses.

3. SOAP - simple object access protocol - a protocol for exchanging structured information during the implementation of web services in computer networks. SOAP uses XML for authentication, authorization, and process interaction across operating systems and allows clients to invoke Web services and receive responses regardless of language and platform.

4. REST API - Representational State Transfer (REST) is an architectural style that implements client and server independently of each other.

3.3 Conclusions

So, in this section, a method for improving the security of the cyber-physical system "Smart Parking" was proposed. The method is to add middleware to improve security to the server part of the cyber-physical system for smart parking.

Also, a scheme of operation of the components of the cyber-physical system "Smart parking" was developed, taking into account the intermediate software, the principle of operation of the intermediate software to improve the security of the cyber-physical system "Smart Parking" was proposed. The architecture of the cyber-physical system for smart parking using middleware was also presented in this section.

CONCLUSIONS

At the moment, the urgent task is to develop methods and means to improve the security of the smart parking software system. The purpose of this work was to analyze the security requirements of the smart parking system software in order to identify the parts of the software that are most vulnerable to external threats, and to develop methods and tools to improve their security.

The thesis proposes a method for improving the safety of the smart parking software system based on the integration of middleware into the software architecture of the smart parking system. The proposed method takes into account all the security criteria of the smart parking system software, i.e. parameters of secure database access, client program security, server security and security API and provides a comprehensive solution to improve the security of the smart parking software system. Using middleware security, it is much easier to determine if the request was actually sent from the native smart parking client and check if it is malicious and does not contain suspicious code. In addition, middleware security software will reduce application uptime if the request is irrelevant or malicious.

The practical value of the obtained results lies in the development of a subsystem for ensuring the security of the cyber-physical system for smart parking.

On the topic of the thesis, she took part in the All-Ukrainian Scientific and Practical Conference Information Technologies and Engineering (IT&I-2023), m. Mykolaiv and published abstracts in the collections of the conference.

LIST OF REFERENCE SOURCES

1. Smart parking in Kyiv URL: https://www.ukrinform.ua/rubric-kyiv/2283219-rozumne-parkuvanna-u-kievi-stali-do-roboti-persi-10-inspektoriv.html (last accessed:12.03.2023)

2. Pavlova O., Kovalenko V., Hovorushchenko T., Avsiyevych V. Neural network based image recognition method for smart parking. *Comput. Syst. Inf. Technol.* 3, 2021. pp. 49–55.

3. Radiuk P., Pavlova O., El Bouhissi H., Avsiyevych V., KovalenkoV. Convolutional Neural Network for Parking Slots Detection. *CEUR Workshop Proceedings*, 2022. 3156, pp. 284–293.

4. Avsiyevych V., Kovalenko V. Cyber-physical system for smart parking based on computer vision technology Black Sea Science 2022. *Proceedings of the International Competition of Student Scientific Works*. Odesa National University of Technology. Odessa: ONUT. 2022. pp. 335-346.

5. SoftServe started testing machine learning-based smart parking. zaxid.net. URL: https://zaxid.net/lvivska_kompaniya_softserve_pochala_testuvannya_parking_sistemi_na_bazi_mashinnogo_navchannya_n1471000(last accessed: 30.11.2022).

6. Parking in Bukovel. URL:https://transferdokarpat.com.ua/articles/bukovel-vartist-poslugiparkingiv (last accessed: 30.11.2022).

7. A startup on smart parking implementation started in Kyiv. URL:https://kyivcity.gov.ua/news/u_kiyevi_startuvav_pilot_iz_vprovadzhennya_rozumno_sistemi_parkuvannya/ (last accessed: 30.11.2022).

8. Інтелектуальна система парковки Acer. URL: https://www.acer.com/ac/ru/RU/content/acerdesign-smart-parking (дата звернення: 30.11.2022).

9. Apple App Store URL:https://www.apple.com/ua/app-store/ (дата звернення: 30.11.2022).

10. Google Play Market URL:https://play.google.com/store (дата звернення: x

11. Gollapudi S. Learn computer vision using opencv: with deep learning CNNs and RNNs : eBook. Berkeley, CA : Apress. 2019. 171 p.

12. Balaji G.V., Bharath K., Nithin S., Pranesh D.M., Shilpa S.B. Object detection using OpenCV and deep learning. *International Journal for Research in Applied Science and Engineering Technology*. 2021. Vol. 9, No. 1. pp. 3920-3923.

13. Parking Startups Are Cashing In on America's Traffic Surge URL:https://www.bloomberg.com/news/articles/2021-07-22/parking-startups-cash-in-on-america-s-post-pandemic-traffic-surge-with-apps (дата звернення: 07.12.2022)

14. Vision AI. Google Cloud. URL: https://cloud.google.com/vision (дата звернення: 02.12.2022)

15. Learn OpenCVC++ in 4 hours URL: https://www.youtube.com/watch?v=2FYm3GOonhk (дата звернення: 02.12.2022)

16. Amato G., Carrara F., Falchi F., Gennaro C., Vairo C. CNRPark+EXT. A Dataset for Visual Occupancy Detection of Parking Lots. URL:http://cnrpark.it/ (дата звернення: 07.12.2022)

17. Parking Lot Database URL: https://web.inf.ufpr.br/vri/databases/parking-lot-database/ (дата звернення: 07.12.2022)

18. Melbourne University Dataset URL: https://melbourne.figshare.com/articles/dataset/MATLABCodeCNNSVM_zip/12978932/1?file=24726374 (дата звернення: 07.12.2022)

19. Marek M. Official repository for the "Image-Based Parking Space Occupancy Classification: Dataset and Baseline" paper. URL: https://github.com/martin-marek/parking-space-occupancy (дата звернення: 07.12.2022)

20. Detectron2 - a platform for object detection, segmentation and other visual recognition tasks. URL:https://github.com/facebookresearch/detectron2 (дата звернення: 07.12.2022)

21. IP-camera URL:https://rozetka.com.ua/hikvision_ds_2cd2121g0_is_c_2_8_mm/p322796167/characteristics/ (last accessed:: 07.12.2022)

22. IP-camera Dahua DH IPC HDW2431 URL: https://rozetka.com.ua/dahua_dh_ipc_hdw2431tp_as_s2_3_6_mm/p175174490/characteristics/ (last accessed: 07.12.2022)

23. Introduction to OpenCV-Python Tutorials URL: https://docs.opencv.org/3.4/d0/de3/tutorial_py_intro.html (дата звернення: 19.02.2023)

24. How to Build an Effective API Security Strategy URL:https://www.gartner.com/en/documents/3834704 (дата звернення: 19.02.2023)

25. The PHP Framework for Web Artisans URL:https://laravel.com/ (дата звернення: 19.02.2023)

26. Laravel Framework GitHub URL: https://github.com/laravel/framework (дата звернення: 19.02.2023)

27. Laravel - Overview URL: https://www.tutorialspoint.com/laravel/laravel_overview.htm (дата звернення: 19.02.2023)

28. Hovorushchenko T., Pavlova O., Kostiuk M. Method of Increasing the Security of Smart Parking System. *JCSANDM*. vol. 12, no. 03, 2023. pp. 297–314.

29. Hovorushchenko T., Boyarchuk A., Pavlova O., Bobrovnikova K. Agent-Oriented Information Technology for Assessing the Initial Stages of the Software Life Cycle. ICTERI Workshops. 2019. pp. 617-632.

30. Understanding OWASP Mobile Top 10 Risks with Real-world Cases. URL: https://appinventiv.com/blog/owasp-mobile-top-10-real-world-cases/amp (дата звернення: 19.02.2023)

31. The top API security risks and how to mitigate them. URL: https://appinventiv.com/blog/how-to-mitigate-api-security-risks/ (дата звернення: 19.02.2023)

32. Lopatto I., Hovorushchenko T., Popov P., Pavlova O. Intelligent Multi-Agent System for Improving the Quality of Software by Taking into Account the Information of the Subject Area at All Stages of its Development. *Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. IDAACS 2021. 2021. 1. pp. 548–551.

33. Amiri W. A., Baza M., Banawan K., Mahmoud M., Alasmary W., Akkaya K. Towards Secure Smart Parking System Using Blockchain Technology. 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC). 2020. pp. 1-2. doi: 10.1109/CCNC46108.2020.9045674.

34. Waheed P., Krishna P.V. Comparing Biometric and Blockchain Security Mechanisms in Smart Parking System. *2020 International Conference on Inventive Computation Technologies* (ICICT). 2020. pp. 634-638. doi: 10.1109/ICICT48043.2020.9112483.

35. Kumar M., Khan M. H., Umar M. S. Smart parking system using RFID and GSM technology. *2017 International Conference on Multimedia, Signal Processing and Communication Technologies* (IMPACT), 2017. pp. 180-184. doi: 10.1109/MSPCT.2017.8364000.

36. Lee C., Leng FTJ., Habeeb RAA., Amanullah MAA, Rehman M., Edge computing-enabled secure and energy-efficient smart parking. *A review, Microprocessors and Microsystems*. Volume 93, 2022

37. Ahmed S., Soaibuzzaman M., Rahman S., Rahaman S. A Blockchain-Based Architecture for Integrated Smart Parking Systems. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops* (PerCom Workshops). 2019, pp. 177-182, doi: 10.1109/PERCOMW.2019.8730772.

38. Biyik M., Allam M., Pieri G., Moroni G., O'Fraifer M., O'Connell M., Olariu M., Khalid M. Smart Parking Systems: Reviewing the Literature, Architecture and *Ways Forward. Smart Cities*. 2021. 4(2). pp. 623-642.

39. Imbugwa G.B., Mazzara M. Towards a Secure Smart Parking Solution for Business Entities. *Advanced Information Networking and Applications*. AINA 2021. Lecture Notes in Networks and Systems. vol 227. Springer. pp. 469-478.

40. Abdulkader O., Bamhdi A. M., Thayananthan A., Jambi K., Alrasheedi A. A novel and secure smart parking management system (SPMS) based on integration of WSN, RFID, and IoT. *2018 15th Learning and Technology Conference (L&T)*. 2018. pp. 102-106, doi: 10.1109/LT.2018.8368492.

41. Singh SK, Pan Y, Hyuk J. Blockchain-enabled Secure Framework for Energy-Efficient Smart Parking in Sustainable City Environment. *Sustainable Cities and Society*. vol. 76. 2022.

42. Garcia M., Rose P., Sung P., El-Tawab E., Secure Smart Parking at James Madison University via the Cloud Environment (SPACE). *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS),* 2016. pp. 271-276. doi: 10.1109/SIEDS.2016.7489313.

43. Anwar A., Ijaz-ul-Haq N., Saadati P. Smart Parking: Novel Framework of Secure Smart Parking Solution using 5G Technology. *2021 IEEE International Smart Cities Conference* (ISC2). 2021, pp. 1-4. doi: 10.1109/ISC253183.2021.9562776.

44. Hovorushchenko T., Pavlova O., Avsiyevych V. Method of Assessing the Impact of External Factors on Geopositioning System Operation Using Android GPS API. *2021 International Scientific and Technical Conference on Computer Sciences and Information Technologies* (CSIT), 2021, 1. pp. 295–298

45. Waheed A., Krishna P. V., Gitanjali J., Sadoun B., Obaidat M. Learning automata and reservation based secure smart parking system. *Methodology and simulation analysis, Simulation Modelling Practice and Theory*. vol. 106. 2021.

46. Atif Y., Ding J., Jeusfeld MA., Internet of Things Approach to Cloud-based Smart Car Parking. *Procedia Computer Science*. vol. 98. 2016

47. Hakim I.M., Christover M., Jaya Marindra A.M. Implementation of an image processing based smart parking system using Haar-Cascade method. *2019 IEEE 9th Symposium on Computer Applications Industrial Electronics* (ISCAIE-2019). pp.

222–227. IEEE Inc., Penang, Malaysia, 27-28 April 2019. DOI:10.1109/ISCAIE.2019.8743906

48. Manjula G., Rajulu G.G., Anand, J.T. Thirukrishna. Implementation of smart parking application using IoT and machine learning algorithms. *Computer Networks and Inventive Communication Technologies*. Springer Singapore, Singapore. 2022. pp. 247–257. doi:10.1007/978-981-16-3728-5_18.

49. Vakula D., Kolli Y.K.. Low cost smart parking system for smart cities. *International Conference on Intelligent Sustainable Systems*, 2017. DOI: 10.1109/ISS1.2017.8389415.

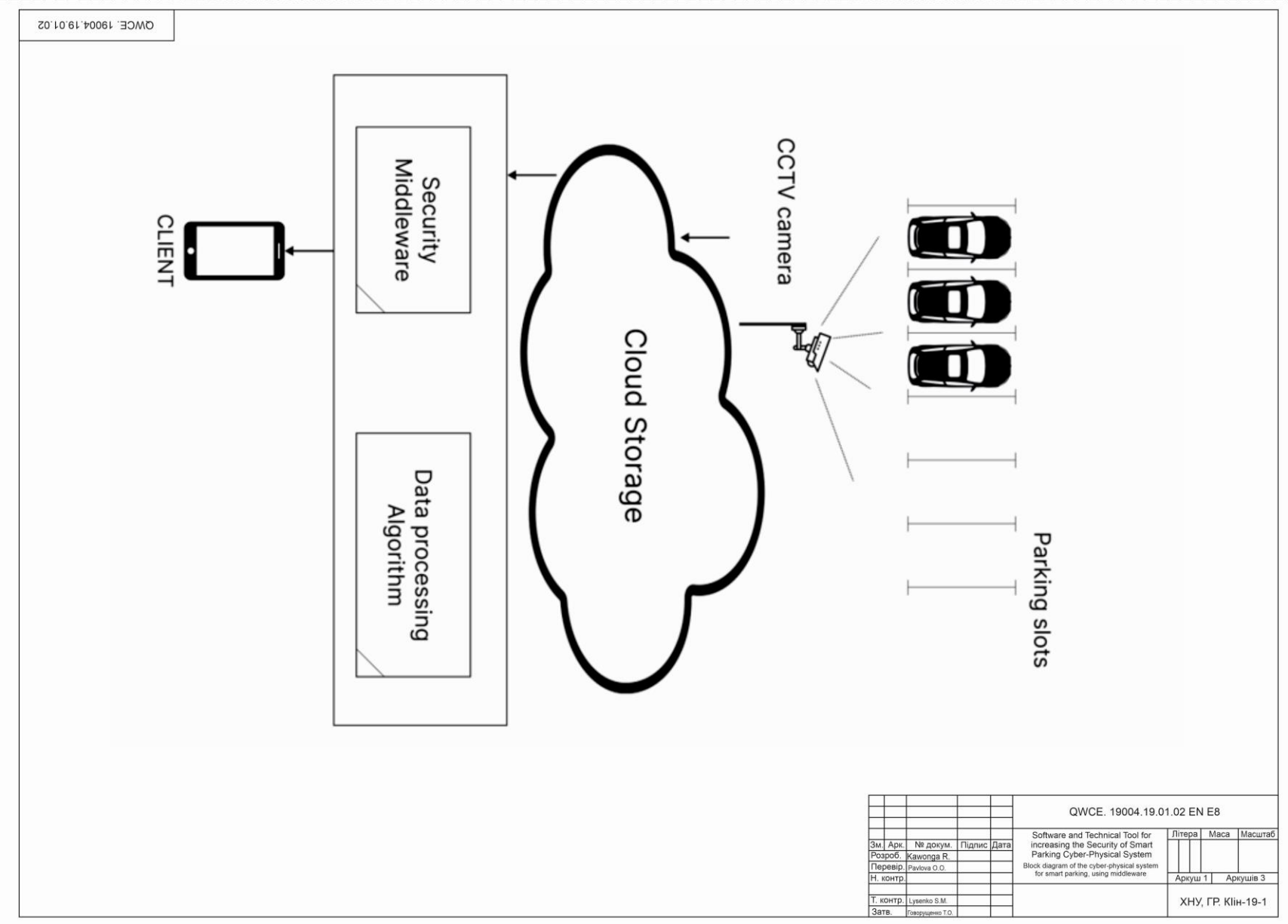
50. Tkachenko R., Izonin I., Dronyuk I., Logoyda M., Tkachenko P. Recovery of missing sensor data with grnn-based cascade scheme. *International Journal of Sensors, Wireless Communications and Control*. 2021, vol. 11, no.5. pp. 531–541

51. Zaitseva E., Levashenko V. Construction of a reliability structure function based onuncertain data. IEEE *Transactions on Reliability*. vol. 65, no. 4. 2016. pp. 1710 – 1723.

52. Zaitseva E., Levashenko V. Reliability analysis of multi-state system with application of multiple-valued logic. *International Journal of Quality and Reliability Management*. 2017. vol. 34, no. 6. P.862 – 878.

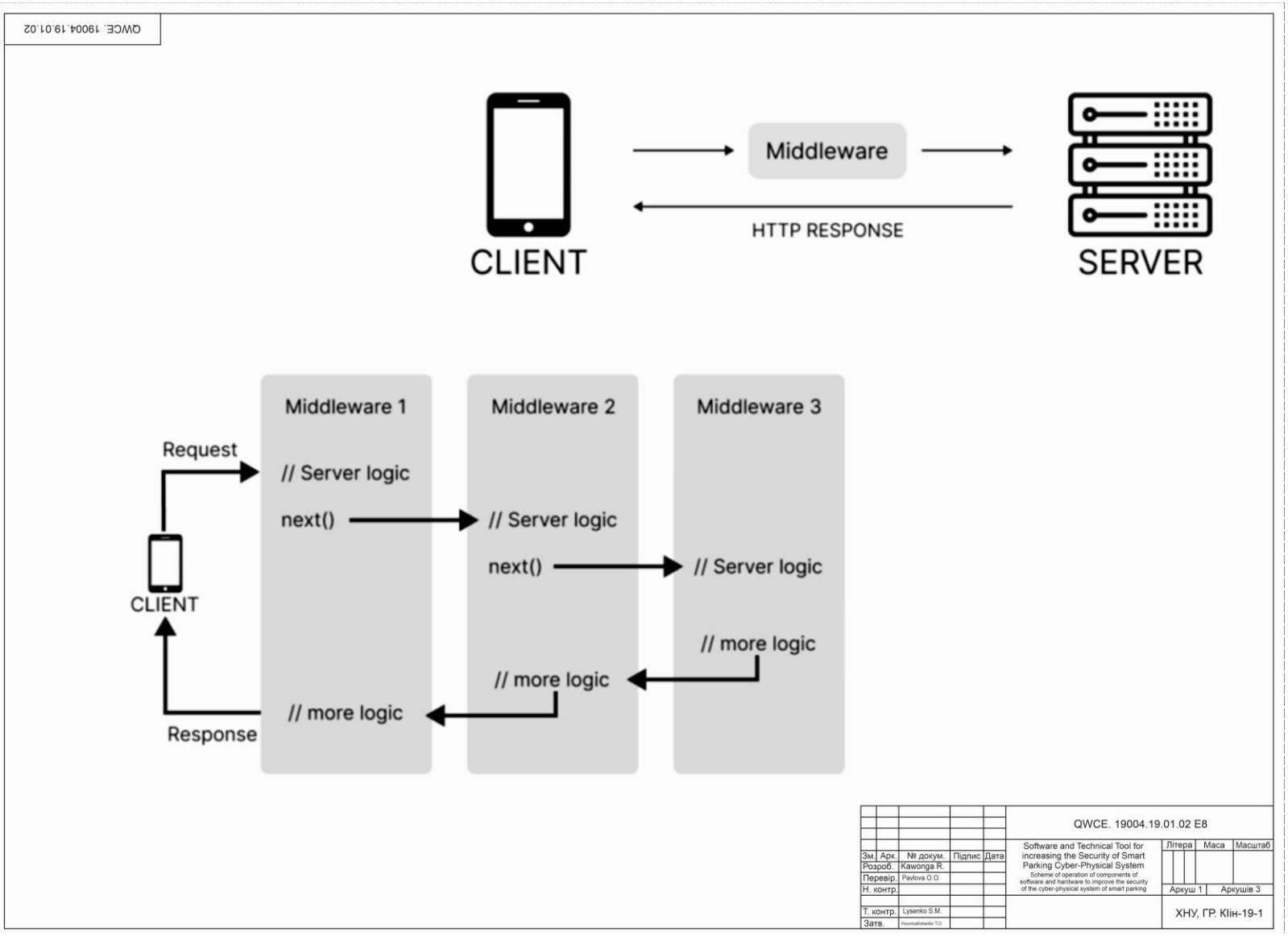
Appendix A

Copy of the drawing "Block diagram of the cyber-physical system for smart parking, using middleware"



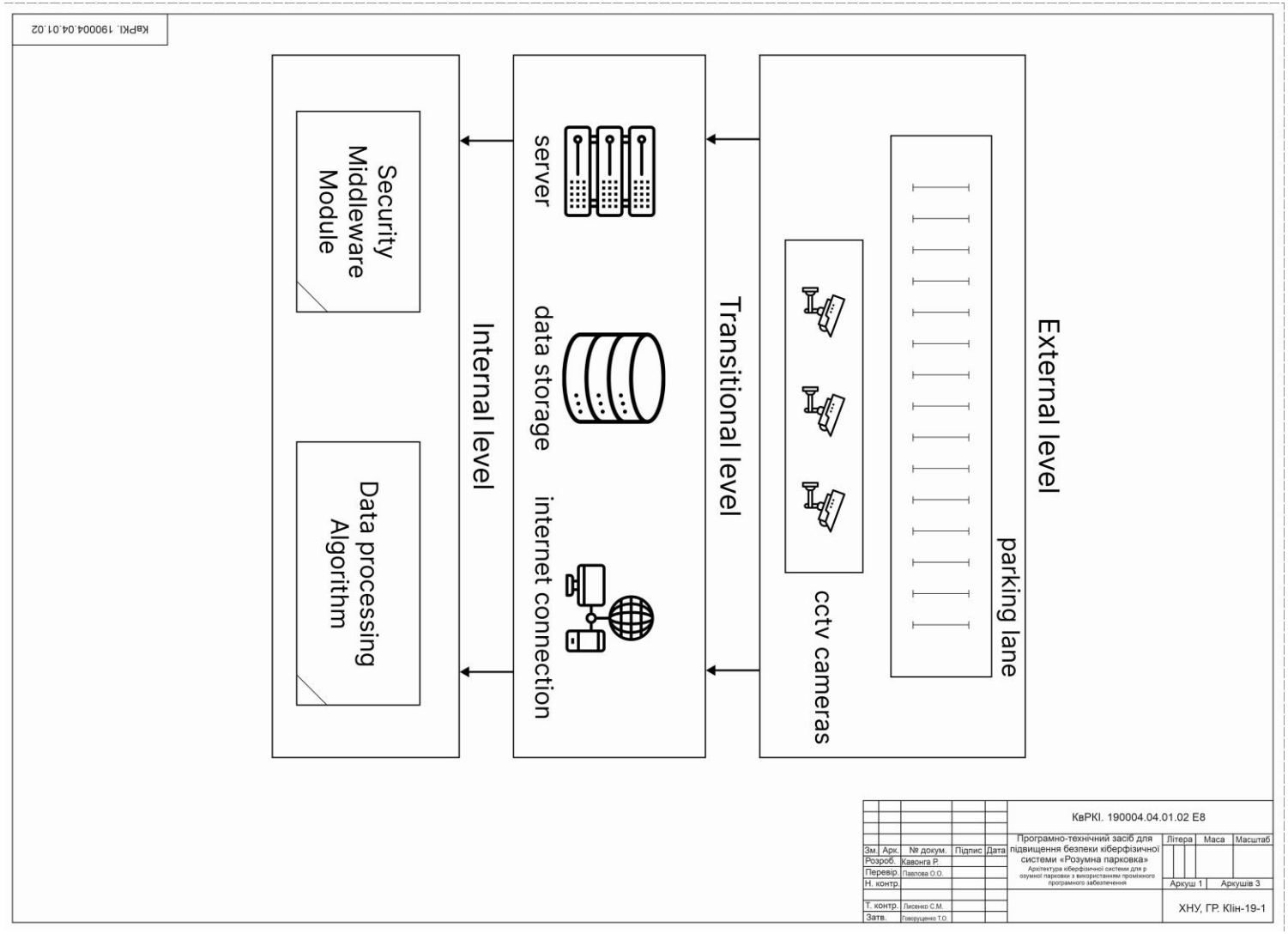
Appendix B

Copy of the drawing "Scheme of operation of components of software and hardware to improve the security of the cyber-physical system of smart parking"



Appendix C

Copy of the drawing "Architecture of the cyber-physical system for smart parking using middleware"



To the head of the department of KIIS
Doctor of Technical Sciences, Prof.
Hovorushchenko T. O.

Kawonga Rose
Full name of the applicant of higher education
FIT, 4th year, group KI2iн-19-1

STATEMENT

With the rules of the current Regulation "On the system of ensuring academic integrity at the Khmelnytskyi National University" dated 01.07.2022, according to which the detection of plagiarism is a reason for refusing to admit a qualification work to the defense and applying measures of disciplinary and academic responsibility, (a) is familiar. I have been informed about the use of software and technical tools to check the qualification works of higher education applicants for plagiarism and I give my consent to the university processing and saving my work in the university's institutional repository.

I also grant the university the right to transfer my work for processing and storage in the databases of software and technical tools (Unicheck and Anti-Plagiarism) and to use the work to detect plagiarism in other works that are checked by software and technical tools and users who have access to these software and technical means, exclusively for limited purposes to detect plagiarism in the texts of works.

Work for review by the university is provided in printed and electronic versions. The electronic version of my work coincides (identical) with the printed one.

April 22, 2023

Signature

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічний засіб для підвищення безпеки кіберфізичної системи "Розумна парковка"

Автор: Кавонга Роуз

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Павлова Ольга Олександрівна, д.ф, ст.викладач

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 2,38% і адресується до 918 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи О. О. Павлова

Гарант ОП Є. Г. Лисенко

Завідувач кафедри КПС Т. О. Говорущенко

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Кавонга Роуз

Тема: Програмно-технічний засіб для підвищення безпеки кіберфізичної системи «Розумна парковка»

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень ___3___ Кількість сторінок записки ___64___

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є підвищення доступності громадських місць для людей з обмеженими можливостями.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проведено аналіз існуючих рішень, методів та підходів до реалізації програмно-технічних засобів для підвищення безпеки кіберфізичних систем для розумної парковки. В другому розділі кваліфікаційної роботи виконано обґрунтування вибору компонентів та середовища реалізації, а саме: апаратне середовище, функційні та нефункційні вимоги до розроблюваного програмно-технічного засобу та програмне середовище для реалізації. В третьому розділі кваліфікаційної роботи розроблено структурну схему та алгоритм роботи програмно-технічного засобу для підвищення безпеки кіберфізичної системи для розумної парковки.

4. Позитивні сторони роботи: висока актуальність та практична цінність роботи.

5. Негативні сторони роботи: недостатня увага приділена реалізації розроблюваної системи.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному інженерно-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Мартинюк Валерій Володимирович, д.т.н., професор, завідувач кафедри автоматизації та комп'ютерно-інтегрованих технологій Хмельницького національного університету

"5" 06 2023 р. _____(підпис)

Mon May 29 06:31:36 EEST 2023, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 25%**

ID: 114198 Назва: Software and Technical Tool for increasing the Security of Smart Parking Cyber-Physical System Додано в БД: 2023-05-29 Автора: R. Kawonga Керівники: O.O. Pavlova Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	74206	616	963 (1%)	11 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK

Ім'я користувача:
Кафедра КІ

ID перевірки:
1015295471

Дата перевірки:
29.05.2023 07:26:29 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
29.05.2023 07:47:46 EEST

ID користувача:
100005591

Назва документа: **Kawonga_Software and Technical Tool for increasing the Security of Smart Parking Cyber-Physical System**

Кількість сторінок: **55** Кількість слів: **12745** Кількість символів: **86491** Розмір файлу: **1.76 MB** ID файлу: **1014967444**

2.38%
Схожість

Найбільша схожість: **0.23%** з Інтернет-джерелом (http://journals.khnu.km.ua/vestnik/wp-content/uploads/2021/07/2021..

2.19% Джерела з Інтернету	206	Сторінка 57
0.51% Джерела з Бібліотеки	6	Сторінка 58

1.45% Цитат

Цитати	7	Сторінка 59
Посилання	1	Сторінка 59

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	1

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-технічний засіб для підвищення безпеки кіберфізичної системи «Розумна парковка»
Назва теми

КвРКІ. 19004.04.01.02 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»
Назва

Виконав: студент IV курсу, група КІін–19–1 _____ Підпис Р.Кавонга Ініціали, прізвище

Керівник _____ Підпис, дата О.О. Павлова Ініціали, прізвище

Нормоконтролер _____ Підпис, дата С.М. Лисенко Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної інженерії та інформаційних систем _____ Підпис Т.О. Говорущенко Ініціали, прізвище

«5» червня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЯ ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

"11" 01 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Кавонги Роуз

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб для підвищення безпеки кіберфізичної системи «Розумна парковка»

Керівник проекту (роботи) Павлова О.О., д.ф., ст.викл.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 09.06.2023 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Огляд існуючих систем вирішення проблеми

Проблема безпеки кіберфізичної системи для розумного паркування

Впровадження програмно-технічного засобу для підвищення безпеки кіберфізичної системи "Розумна Парковка"

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Структурна схема кіберфізичної системи для розумного паркування, із застосуванням проміжного програмного забезпечення

Схема роботи компонентів програмно-технічного засобу для підвищення безпеки кіберфізичної системи розумної парковки

Архітектура кіберфізичної системи для розумної парковки з використанням проміжного програмного забезпечення

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІІС		
Антиплагіат	Нічепорук А.О., доцент кафедри КІІС		

7. Дата видачі завдання «_11_» 01 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2023	виконано
3	Робота над розділом 1 – Огляд існуючих систем вирішення проблеми	01.03.2023	виконано
4	Робота над розділом 2 – Проблема безпеки кіберфізичної системи для розумного паркування	01.04.2023	виконано
5	Робота над розділом 3 – Впровадження програмно-технічного засобу для підвищення безпеки кіберфізичної системи "Розумна Парковка"	30.04.2023	виконано
6	Оформлення пояснювальної записки згідно вимог	21.05.2023	виконано
7	Попередній захист ВКР	26.05.2023	виконано
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент _____ Р. Кавонга
Підпис Ініціали, прізвище

Керівник проекту (роботи) _____ О. О. Павлова
Підпис Ініціали, прізвище

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 19004.04.01.02 ПЗ	Пояснювальна записка	65		
			Графічні матеріали			
2		КвРКІ 19004.04.01.02 Е8	Структурна схема кіберфізичної системи для розумного паркування, із застосуванням проміжного програмного забезпечення	1		
3		КвРКІ 19004.04.01.02 Е8	Схема роботи компонентів програмно-технічного засобу для підвищення безпеки кіберфізичної системи розумної парковки	1		
		КвРКІ 19004.04.01.02 Е8	Архітектура кіберфізичної системи для розумної парковки з використанням проміжного програмного забезпечення			
					1	
4						

КвРКІ 19004.04.01.02 ВП

Зм	Арк	№ докум	Підпис	Дата
Розробив		Кавоига		05.06
Перевір.		Павлова		05.06
Н. контр.		Лисенко		05.06
Затв.		Говорущенко		05.06

Відомість проекту

Літера	Аркуш	Аркушів
У	1	1

ХНУ, КІін-19-1

АНОТАЦІЯ

Тема кваліфікаційної роботи: Програмно-технічний засіб для підвищення безпеки кіберфізичної системи “Розумна парковка”.

Автор роботи: Кавонга Роуз.

Керівник роботи: Павлова Ольга Олександрівна.

Пояснювальна записка: 65 с., 15 рис., 5 табл., 3 дод., 65 джерел.

Графічна частина: 3 плакати.

РОЗУМНА ПАРКОВКА, КІБЕРФІЗИЧНА СИСТЕМА, СЕРВЕРНА ПІДСИСТЕМА, КАМЕРА ЗОВНІШНЬОГО СПОСТЕРЕЖЕННЯ, СТРУКТУРНА СХЕМА, АРХІТЕКТУРА КІБЕРФІЗИЧНОЇ СИСТЕМИ

Метою кваліфікаційної роботи є підвищення безпеки кіберфізичної системи “Розумна парковка”.

Об’єктом дослідження є процес підвищення безпеки кіберфізичної системи “Розумна парковка”.

Предметом дослідження є програмно-технічний засіб для підвищення безпеки кіберфізичної системи “Розумна парковка”.

Для досягнення визначеної мети застосовуються різноманітні дослідницькі методи, такі як синтез, аналіз та моделювання процесів, принципи системного аналізу, а також підходи, що базуються на теорії множин.

05.06.2023

Дата

Підпис студента

ЗМІСТ

ВСТУП 5
1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ВИРІШЕННЯ ПРОБЛЕМИ 7
1.1 Програмно-технічні засоби для розумного паркування в Україні та світі 7
1.2 Аналіз існуючих методів для забезпечення безпеки систем для розумного паркування 17
1.3 Аналіз програмних рішень для забезпечення безпеки системи для розумного паркування 20
1.4 Висновки 25
2.1 Клієнт-серверна архітектура та можливі фактори загрози безпеці кіберфізичної системи 26
2.1.1 Ризики безпеки на стороні клієнта 27
2.1.2 Ризики безпеки на стороні сервера 31
2.2 Ризики безпеки API 33
2.3 Вибір методів і середовища реалізації програмного забезпечення 35
2.4 Висновки 40
3 ВПРОВАДЖЕННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ ДЛЯ ПІДВИЩЕННЯ БЕЗПЕКИ КІБЕРФІЗИЧНОЇ СИСТЕМИ “РОЗУМНА ПАРКОВКА” 41
3.1 Запропонований спосіб підвищення безпеки кіберфізичної системи “Розумна Парковка” 41
3.2 Структурна схема та алгоритм роботи програмно-технічного засобу підвищення безпеки кіберфізичної системи “Розумна Парковка” 47
3.3 Висновки 61
ВИСНОВКИ 62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 63

ДОДАТОК А 61
ДОДАТОК Б 63
ДОДАТОК В 64

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ПЗ – програмне забезпечення

ПС – програмна система

ОС – операційна система

HTTP – HyperText Transfer Protocol – Гіпертекстовий протокол передачі даних

HTTPS – HyperText Transfer Protocol Secure – Безпечний гіпертекстовий протокол передачі даних

RFID – Radio Frequency Identification – електромагнітний датчик

API – Applied Program Interface – прикладний програмний інтерфейс

ВСТУП

На сучасному етапі розвитку інформаційно-комп'ютерних технологій особливу увагу необхідно приділяти питанням безпеки при розробці програмного забезпечення. Це особливо важливо для критичного програмного забезпечення та програмного забезпечення кіберфізичних систем, оскільки втрата даних або несправності можуть мати непередбачувані, а іноді й критичні наслідки. Некоректна робота алгоритму або помилки в розпізнаванні зображень штучною нейронною мережею можуть призвести до надання некоректного результату. Оскільки клієнт-серверна архітектура є особливо вразливою до різного роду зовнішніх загроз, доцільно передбачити методи та алгоритми захисту та безпеки системи розумного паркування на ранніх етапах життєвого циклу, тобто на етапі проектування архітектури програмного забезпечення. . Це надзвичайно важливо, оскільки вартість виправлення помилок зростає з кожним етапом життєвого циклу.

Комп'ютерне обладнання (наприклад, смартфони, бездротові датчики та персональні комп'ютери) стає дедалі меншим, дешевшим і потужнішим як технічна основа розумних систем паркування. У результаті мобільні та повсюдні обчислення швидко стають важливими компонентами розосереджених мережевих обчислювальних інфраструктур. Він пропонує нам потужну платформу для обчислення інформації в реальному часі з фізичного світу (фізичний компонент) і спілкування з людьми (кіберчастина). Ми не повинні недооцінювати важливість кіберфізичних систем у повсякденному житті.

Актуальність роботи полягає у розробці підсистеми для забезпечення безпеки кіберфізичної системи розумної парковки. Метою дипломної роботи є підвищення безпеки кіберфізичної системи для розумного паркування.

Поставлена мета досягається розв'язанням таких основних задач:

1) провести огляд існуючих рішень та систем забезпечення безпеки для розумних парковок;

2) виконати вибір компонентів та середовища для реалізації задачі;

3) розробити підсистему для забезпечення безпеки кіберфізичної системи для розумного паркування.

Об’єктом дослідження є процес розробки підсистеми для забезпечення безпеки кіберфізичної системи для розумного паркування. Предметом дослідження є підсистема для забезпечення безпеки кіберфізичної системи для розумного паркування.

Практична цінність отриманих результатів полягає у розробці підсистеми для забезпечення безпеки кіберфізичної системи для розумного паркування.

За темою дипломної роботи було взято участь у Всеукраїнській науково-практичній конференції Інформаційні технології та інженерія (IT&I-2023), м. Миколаїв та опубліковано тези у збірниках конференції.

1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ВИРІШЕННЯ ПРОБЛЕМИ

1.1 Програмно-технічні засоби для розумного паркування в Україні та світі

Проблема з вільними паркувальними місцями існує як в Україні, так і в багатьох інших країнах світу. Було розглянуто основні проблеми,пов'язані з паркуванням. Результати аналізу наведені у таблиці 1.1.

Таблиця 1.1 – Результати аналізу проблем з вільними паркомісцями в Україні та світ

Проблема	Опис
Недостатня кількість паркувальних місць	У багатьох містах і населених пунктах не вистачає парковочних місць, щоб задовольнити потреби мешканців, відвідувачів і бізнесу. Це приводить до пошуку водіями вільних місць і паркування на тротуарах, газонах або в заборонених місцях.
Несанкціоноване паркування	Багато водіїв паркують свої автомобілі у заборонених місцях, таких як пішохідні переходи, виїзди із дворів, проїзди для екстреної допомоги, узбіччя доріг тощо. Це створює проблеми для безпеки дорожнього руху, обмежує доступ інших водіїв і перешкоджає нормальному функціонуванню міста.

Продовження таблиці 1.1 – Результати аналізу проблем з вільними паркомісцями в Україні та світ

Висока вартість паркування	У деяких містах паркування може бути дуже дорогим, особливо в центральних районах або на комерційних територіях. Це може бути фінансово вимогливо для водіїв і створює нерівність у доступі до паркувальних місць.
Неефективне використання наявних місць	Часто виявляється, що існуючі парковочні місця не використовуються ефективно. Наприклад, деякі автомобілі можуть займати місця протягом тривалого часу, навіть коли вони не використовуються, а інші водії не можуть знайти вільні місця.
Недостатня інфраструктура	Відсутність адекватної парковочної інфраструктури, такої як парковки біля громадських будівель, торгових центрів, станцій громадського транспорту тощо, ускладнює пошук водіями місць для паркування.

Для вирішення цих проблем уряди і муніципалітети можуть вживати такі заходи, як:

1. Розширення парковочної інфраструктури шляхом будівництва нових парковок і багатоповерхових паркінгів.

2. Впровадження електронних систем керування паркуванням, які дозволяють водіям знаходити вільні місця та оплачувати паркування за допомогою мобільних додатків або інтернету.

3. Застосування технологій "розумного паркування", які використовують датчики і системи керування для оптимізації використання парковочних місць і покращення ефективності паркування.

4. Законодавчі заходи, які передбачають введення штрафів за паркування у заборонених місцях і неправильну парковку.

5. Сприяння використанню альтернативних видів транспорту, таких як велосипеди, електротранспорт, громадський транспорт, що зменшить потребу в паркувальних місцях для автомобілів.

Розуміння цих проблем і прийняття належних заходів можуть сприяти поліпшенню ситуації з паркуванням як в Україні, так і в інших країнах.

Було проведено дослідження щодо вирішення проблеми безпеки системи розумного паркування з використанням різних методологій та інструментів. З точки зору забезпечення системи безпеки програмного забезпечення Smart Parking, основними критеріями, яких необхідно дотримуватися при створенні системи безпеки для розумного паркування, є апаратна безпека, безпека апаратно-програмного з’єднання та безпека програмного забезпечення.

Можна виділити такі вимоги:

5. перевірка параметрів безпечного доступу до бази даних;

6. безпека клієнтської програми;

7. безпека сервера;

8. безпека API, якщо дизайн програмної системи розумного паркування підтримує його використання;

Основний принцип роботи систем для розумного паркування полягає в тому, що рішення щодо розповсюдження інформації про дорожній рух має

прийматися інфраструктурою, а не окремими автомобілями, які можуть мати неточні або неповні відомості про маршрут. Інфраструктура у такій системі досягається шляхом встановлення ременів датчиків через рівні проміжки в дорогу. Кожен ремінь складається з кількох п'єзоелектричних детекторів тиску, рудиментарного агрегату та термоядерного двигуна, а також кількох невеликих приймачів. Датчики тиску в кожному ремені дозволяють пов'язувати кожне повідомлення з реальним транспортним засобом, що проїжджає через ремінь, усуваючи потребу в індивідуальній ідентифікації автомобілів, уникаючи проблем безпеки. Є дві безпосередні переваги впровадження поясів замість придорожньої інфраструктури. По-перше, ремені значно менш вразливі до маніпуляцій, а по-друге, вони краще розташовані для виявлення автомобілів, що проїжджають повз, і зчеплення з ними простим і безпечним способом. Транспортні засоби обладнані пристроєм захисту від несанкціонованого доступу (TRD). TRD отримує інформацію від усіх вузлів автомобіля, включаючи блок бездротової трансивера, спідометр, показники бензобаку, датчики тиску в шинах і датчики зовнішньої температури. Варто зауважити, що Smart Parking — це інтелектуальна комп'ютерна програма для паркування та революційна інфраструктура, яка забезпечує безпеку та конфіденційність. Для початку автомобілісти на дорозі можуть побачити та забронювати місце для паркування. Паркування може бути ефективною та постійною послугою. По-друге, використовуючи інфраструктуру такої розумної парковки, конфіденційність водіїв враховується та захищається. Нарешті, інформаційна безпека забезпечується використанням поясної інфраструктури та техніки шифрування/дешифрування. Результати моделювання показують, що запропонований підхід призводить до високого використання місця для паркування та короткого часу для пошуку місця для паркування. Кіберфізична система складається з чотирьох модулів:

Модуль драйвера керує зв'язком із апаратними пристроями. Драйвер датчика для ременів, драйвер сигналізації для трансивера ближнього радіусу дії

автомобіля та драйвер IFD для ідентифікації автомобіля є частиною модуля драйвера.

Модуль зв'язку; отримує та надсилає повідомлення між відправником та одержувачем. Цей модуль імітує процес зв’язку та здійснює контроль помилок, наприклад перевірку контрольної суми та виправлення помилок, для зв’язку транспортного засобу з інфраструктурою (V2I). Оскільки час реакції зв’язку обмежений, метою є покращення швидкості зв’язку та точності повідомлення. Цей модуль передає повідомлення між двома фіксованими трансиверами для зв’язку між інфраструктурами (I2I), такими як трансивер на парковці та трансивер у кабінці.

Функціональний модуль; це основна функція системи паркування, яка включає моніторинг, реєстрацію, бронювання та керування рекламою. Функціональний модуль може спілкуватися з апаратними пристроями та передавати/отримувати дані, не знаючи тонкощів нижчих рівнів завдяки своїй підписці на модуль драйвера та модуль зв’язку.

Прикладний модуль; контролює всю систему паркування. Основні функції програми включають адміністрування рахунків (управління готівкою та кредитами/дебетами), керування операціями, відмовостійкість та керування обслуговуванням.

Декомпозиція елементів кіберфізичної системи на основі давачів наведена на рисунку 1.1.

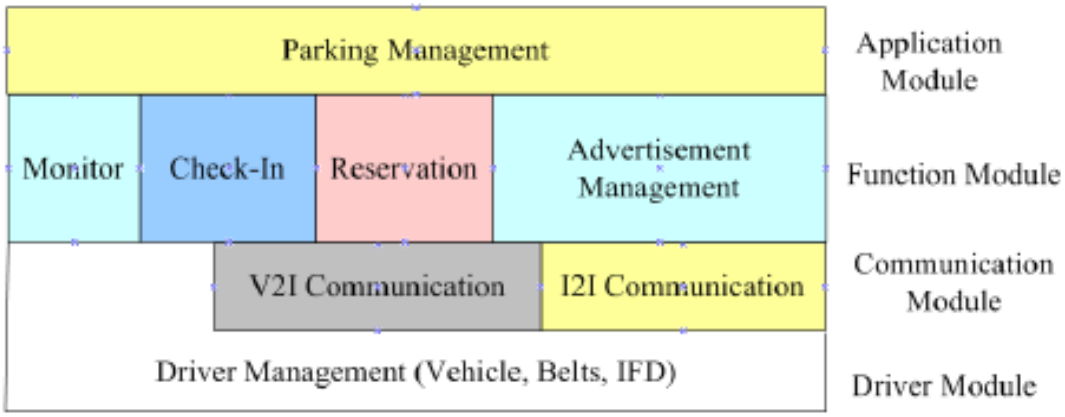


Рисунок 1.1 – Декомпозиція елементів кіберфізичної системи для розумного паркування на основі давачів

Також сьогодні досить популярною технологією для забезпечення безпеки розумної парковки є технологія використання блокового ланцюга - блокчейну.

Постачальник послуг паркування, мережа блокчейн-з'єднань і користувач є трьома дійовими особами в запропонованій системі. В інтегрованій системі провайдер послуг паркування надає паркування як послугу, оновлює місце для паркування. Мережа блокчейну включає публічну книгу, яка оновлюється лише дійсними транзакціями. Для підтвердження транзакцій використовується процес консенсусу. Учасник, який шукає місце для паркування, називається користувачем паркування. Для спілкування з інтегрованою системою розумного паркування кожен учасник має власний інтерфейс програми.

Припустимо, що в місті є багато варіантів інтелектуального автопаркування від різних компаній, що надають послуги паркування. Також припустимо, що для спрощення кожною розумною парковкою керує один постачальник послуг. Кожна стоянка пов'язана з розумною системою паркування на основі блокчейну. Загалом, кожне місце для паркування містить локальну копію книги (тобто локальний блок). У системі є два види транзакцій. Спочатку потрібно врахувати дані датчика паркування. Припустимо, що кожне місце для паркування на розумній автостоянці має пристрій інтернету речей (IoT) (наприклад, датчик паркування), який може генерувати доступність паркування автомобіля як транзакцію.

Для здійснення транзакції кожен постачальник послуг паркування автомобілів має смарт-контракт. Коли статус паркувального місця змінюється з «вільного» на «зайнятий», відповідний пристрій IoT створює транзакцію. Подібним чином, коли місце для паркування змінюється з «зайнятого» на «вільне», пристрій IoT створює транзакцію. Транзакція спочатку надсилається в локальний блок. Транзакція надсилається до мережі блокчейн для перевірки

локальним блоком. По-друге, є інформація про вартість паркування. Припустимо, що постачальники послуг паркування визначають тарифи на паркування на основі часу. Вони розробляють розумні контракти для плати за паркування.

Смарт-контракт ціноутворення на паркування надсилається в мережу блокчейн. Транзакція створюється щоразу, коли ціна паркування динамічно змінюється залежно від часу. Транзакція передається в мережу блокчейн для перевірки. Потім транзакція перевіряється мережею блокчейн за допомогою методу консенсусу. Якщо транзакція правильна, вона реєструється в публічній книзі. В результаті оновлюються всі локальні блоки. Декомпозиція елементів кіберфізичної системи для розумного паркування на основі технології блокчейн наведена на рисунку 1.1.

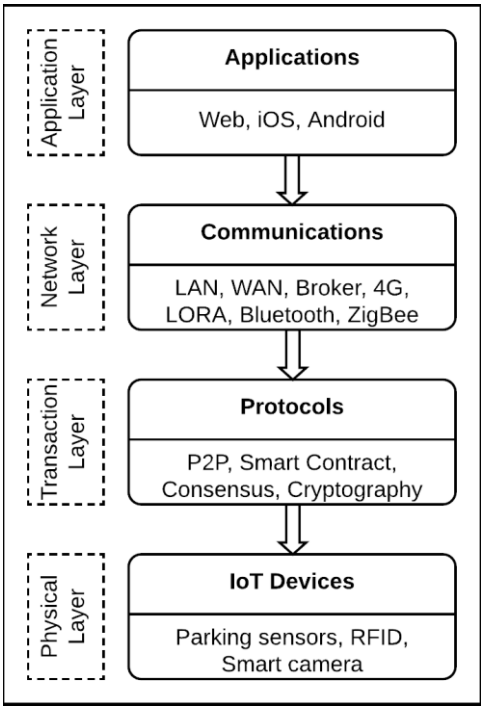


Рисунок 1.2 – Декомпозиція елементів кіберфізичної системи для розумного паркування на основі технології блокчейн

Верхнім рівнем архітектурного стеку є прикладний рівень, який дозволяє учасникам взаємодіяти з системою. Користувачі можуть шукати та бронювати паркувальні місця за допомогою програми для мобільного телефону (наприклад,

Android або iOS) або веб-додатку. Подібним чином постачальники послуг паркування можуть подавати в інтегровану систему інформацію, пов’язану з паркуванням (наприклад, наявність місць для паркування та пропозиції). Користувач підключається до мережі блокчейн через прикладний рівень і може використовувати додаток для надсилання запитів до інтегрованої системи паркування. Інтегрована система відповідає за те, щоб рекомендувати відповідну зону паркування на основі вподобань користувача та наявності. Оскільки користувачі безпосередньо взаємодіють з інтегрованою системою, цей рівень надає кінцевим користувачам найвищий рівень обслуговування.

Мережевий рівень забезпечує зв'язок між засобами паркування, інтегрованою системою та користувачами. Цей рівень надсилатиме дані від користувачів і місць паркування до інтегрованої системи. Цей рівень включатиме різні комунікаційні технології, такі як LAN і WAN, які будуть використовуватися користувачами, постачальниками послуг паркування та пристроями IoT, пов’язаними з системою паркування (наприклад, датчики паркування та камери безпеки). У рамках стандартної пропозиції мережевий рівень безперебійно надає зацікавленим сторонам розподілену публічну книгу та контент-послуги. Він включає багато технологій бездротового зв’язку (наприклад, Lora, Bluetooth, Wi-Fi тощо), а також сучасні технології GSM, такі як 4G і 5G. Цей рівень також забезпечує масштабованість. Наприклад, це дозволяє динамічно додавати та вилучати зацікавлених сторін з інтегрованої системи. Мережевий рівень також забезпечує безпеку фізичного рівня системи.

Рівень транзакцій відповідає за транзакції між вузлами мережі. Це також забезпечить повний процес консенсусу мережі блокчейн. Користувачі та паркувальні об’єкти безпечно обмінюватимуться даними за допомогою розумних контрактів і механізмів консенсусу. Завдяки цьому шару центр паркування також оновлюватиме публічну книгу. Цей рівень з’єднується з основною мережею блокчейну через інтерфейс інтегрованої системи. Цей рівень також перевіряє нові транзакції. Крім того, рівень транзакцій підтримує прозорість транзакцій і

захищає передачу даних за відсутності довіреної третьої сторони. Ми можемо усунути вузькі місця системи та центральні джерела збоїв за допомогою розподіленого дизайну на основі P2P. Крім того, блокчейн розглядатиме передачу даних користувачів як транзакції, які підтверджуватимуться смарт-контрактами. Таким чином, інформація користувачів буде незмінною та розподілятиметься з часом за допомогою криптографічної технології блокчейну.

Фізичний рівень складається з кількох видів пристроїв IoT. Протокол однорангової мережі з'єднує всі ці пристрої в єдину мережу. Основний компонент цього рівня включає в себе різні види датчиків і приводів. Будуть додаткові вбудовані технології, такі як Raspberry Pi та Arduino, на додаток до пристроїв WSN. Рівень транзакцій транспортуватиме дані з пристрою IoT на сервер центру паркування. Потім однорангова мережа підключиться до серверів центру паркування та оновить публічну книгу. Крім того, цей рівень забезпечує відстеження даних датчиків і приводів і відповідальність у всій одноранговій мережі. Оскільки надійність є важливою характеристикою запропонованої нами системи, дані можна безпечно й безпечно транспортувати з пристроїв Інтернету речей за допомогою блокчейну, безпечного незмінного сховища. Доступність певного паркувального місця буде визначатися з фізичного рівня за допомогою датчиків пристроїв IoT. Користувача буде перевірено за допомогою криптографії, а публічна книга буде оновлена інформацією про вільні паркувальні місця. Розумний контракт оброблятиме техніку криптографічної перевірки на рівні транзакцій. Щоб забронювати місце для паркування, користувач може подати запит із прикладного рівня, який буде маршрутизовано через мережевий рівень.

Технологія Smart Car Parking на основі інфрачервоних датчиків дозволяє паркувати транспортні засоби вертикально, поверх за поверхом, зводячи до мінімуму необхідний простір. Система управляється програмним забезпеченням, створеним за допомогою контролера Arduino, мінімізуючи час, який людина витрачає на пошук місця паркування та паркування автомобіля вручну.

Декомпозиція елементів кіберфізичної системи для розумного паркування на основі інфрачервоних датчиків наведена на рисунку 1.3

Для кожного паркувального місця виділено ІЧ-датчик. ІЧ-датчик більше не транслює сигнал вільного місця після того, як автомобіль припарковано, і система знає, які паркувальні місця заповнені. У системі паркування датчик CO2 використовується для контролю та моніторингу вуглекислого газу. Датчик CO2 використовується в системі паркування для керування та вимірювання вуглекислого газу. Коли настає темрява, LDR надає спливаюче повідомлення. РК-дисплей на вході буде регулярно опитувати всі ІЧ-датчики, а також коли автомобіль припаркований і не припаркований, щоб відобразити доступні місця. Коли користувач виявляє будь-які вільні місця на РК-дисплеї, він переходить до датчика зчитувача RFID-міток і пропонує датчику свою картку RFID.

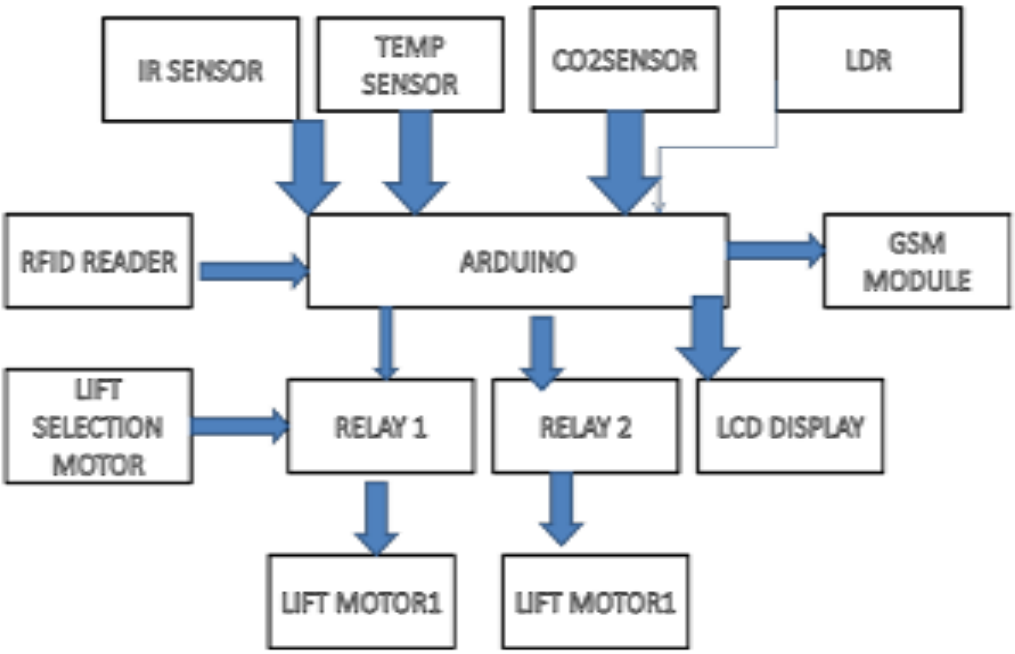


Рисунок 1.3 – Декомпозиція елементів кіберфізичної системи для розумного паркування на основі інфрачервоних датчиків

Потім користувач виходить з автомобіля та встановлює його на вхідну конвеєрну стрічку. Потім автомобіль розміщується на попередньо вказаному паркувальному місці, і кількість на РК-дисплеї збільшується, як і годинник

реального часу для цього місця паркування. Тепер ліфт повернеться на рівень землі або в нейтральне положення, щоб задовольнити потреби інших. RFID-КАРТКА для списання грошей з банківського рахунку користувача. Потім ліфт транспортує транспортний засіб на перший поверх, де користувач може вивезти його з автостоянки. Система також включає в себе контролер моделі GSM SIM 900, який відповідатиме на SMS-запит користувача, надісланий на номер GSM щодо вільних паркувальних місць, шляхом визначення вільних паркувальних місць за допомогою контролера Arduino та ІЧ-датчиків, встановлених на паркувальних місцях. «Доступні паркувальні місця 2, 3 о 10:30 ранку 20.04.2017» або «Наразі немає доступних паркувальних місць» є прикладом сповіщення, яке вказує на доступні паркувальні місця. У результаті система з усіма цими функціями забезпечує точний, ефективний і надійний засіб паркування автомобілів для сучасних міст.

1.2 Аналіз існуючих методів для забезпечення безпеки систем для розумного паркування

Було проведено численні дослідження для вирішення проблеми безпеки системи розумної парковки з використанням різних методів та інструментів. Основними критеріями, якими необхідно керуватися при розробці системи безпеки для розумного паркування, є апаратна безпека, апаратно-програмна безпека підключення та програмна безпека. З точки зору забезпечення безпеки програмної системи Smart Parking можна виділити наступні критерії:

4. перевірка параметрів безпечного доступу до бази даних;
5. безпека клієнтської програми;
6. безпека сервера;

API безпеки, якщо його використання передбачено архітектурою програмної системи розумного паркування.

Проведемо сучасний аналіз відомих рішень і методів, спрямованих на підвищення безпеки. У статті [6] пропонується безпечна розумна система паркування з використанням технології блокчейн, яка використовує техніку маскування для захисту місцезнаходження водіїв.

У [8] надано рішення для запобігання викраденню транспортного засобу на парковці за допомогою технології RFID та GSM. Методи енергозбереження на основі периферійних обчислень та IoT запропоновані в [9].

У [10] представлено інклюзивну, довгострокову, ефективну та добре діючу систему розумного автономного паркування транспортних засобів (SAVP). Автори представляють інтегровану систему розумного паркування, яка об'єднує кількох постачальників послуг паркування в рамках єдиної платформи з метою надання єдиних інформаційних послуг щодо паркування для пасажирів розумного міста.

Основна роль дослідження в роботі [11] полягає в аналізі розумних рішень для паркування з технічної точки зору, підкреслюючи доступні системи та датчики, як зазначено в літературі. Огляд має на меті надати вичерпну інформацію про створення розумних рішень для паркування. Цілісне дослідження поточного стану систем розумного паркування має включати класифікацію таких систем як технології виявлення великих автомобілів.

У статті [12] пропонується система управління паркуванням, орієнтована на суб'єктів господарювання. Запропонована система буде зосереджена на конфіденційності для різних суб'єктів, які використовують систему. Стаття спрямована на вдосконалення вже існуючих досліджень розумного паркування з використанням блокчейну. У цьому документі пропонується система керування паркуванням, яка базуватиметься на JPMorgan Quorum.

Стаття [13] спрямована на розробку безпечної та інтелектуальної системи керування моніторингом паркування (SPMS) на основі інтеграції WSN, RFID та IoT.

Натхненні технологією Blockchain і AI, автори [14] пропонують безпечну структуру з підтримкою Blockchain для енергоефективного розумного паркування в екологічно безпечному міському середовищі.

JMU Secure Smart Parking через хмарне середовище запропоновано в [15]. Використовуючи радіочастотний ідентифікаційний сканер, наша система може підраховувати кількість транспортних засобів, які в'їжджають і виїжджають з кожної автостоянки в кампусі.

Інтелектуальна система паркування в місті на основі мереж мобільного зв'язку 5-го покоління (5G) запропонована в [16]. Технологія мобільного зв'язку 5G має дві важливі переваги: високу швидкість передачі даних і низьку затримку передачі, тому вона може краще задовольнити швидкий розвиток Інтернету речей (IoT).

Стаття [19] представляє програму незавершеної роботи, яка сприяє створенню нових бізнес-рішень і результатам найсучасніших досліджень. Автори розкривають багаторівневу систему PSP-бізнес-моделі через міждисциплінарні дослідницькі блоки, де на кожному рівні очікується отримання оригінальних результатів.

Було проведено аналіз останніх досліджень [6-19] і виділено найбільш часто використовувані методи забезпечення Smart Parking Security. Це: використання технології блокчейн [6, 7, 10, 14], застосування біометричних механізмів безпеки [7], радіочастотна ідентифікація (RFID) та використання бездротової сенсорної мережі (WSN) [8, 13] на основі хмарного середовища. [15, 19], технологія 5G [16], нейронні мережі загальної регресії (GRNN) [23], нечітка логіка та невизначені дані [24], багатозначна логіка [25]. Але всі ці дослідження зосереджені на вирішенні одного-двох критеріїв безпеки Smart Parking System і не забезпечують вирішення всіх вищезазначених критеріїв у комплексі. Отже, з метою підвищення безпеки системи розумної парковки необхідно провести аналіз архітектури програмного забезпечення. Метою такого аналізу є вибір частин програмного забезпечення, які є найбільш уразливими до зовнішніх загроз, і надання рішення

для забезпечення безпеки програмного забезпечення системи розумної парковки з точки зору дотримання всіх критеріїв у комплексі.

Таким чином, актуальним завданням є аналіз вимог до безпеки програмного забезпечення системи розумної парковки з метою виявлення частин програмного забезпечення, які є найбільш уразливими до зовнішніх загроз, і розробки методів і засобів підвищення їх безпеки. Враховуючи вищезазначене, метою даного дослідження є розробка методів підвищення безпеки системи розумної парковки з урахуванням вузьких місць у програмній системі та частин, які є найбільш уразливими до зовнішніх факторів загрози.

1.3 Аналіз програмних рішень для забезпечення безпеки системи для розумного паркування

Системи для розумного паркування стають все більш поширеними, пропонуючи інноваційні рішення для управління паркуванням та покращення його ефективності. Однак, разом зі зростанням популярності таких систем виникають нові виклики щодо безпеки. Забезпечення безпеки в системах розумного паркування вимагає застосування відповідних програмних рішень. У даній статті буде проведено аналіз деяких програмних рішень, які використовуються для забезпечення безпеки в системах розумного паркування.

"Smart Parking Security Suite" є комплексним програмним рішенням, спеціально розробленим для забезпечення безпеки в системах розумного паркування. Він пропонує широкий набір функцій, включаючи відеоспостереження, систему контролю доступу, автоматичне розпізнавання номерних знаків та систему виявлення вторгнень. Це дозволяє виявляти незаконну діяльність, таку як використання підроблених міток для паркування, вандалізм або несанкціонований доступ до парковочних майданчиків.

"Parking Security Management System" - це програмне рішення, яке забезпечує безпеку в системах розумного паркування шляхом інтеграції з різними

пристроями та технологіями. Воно включає систему контролю доступу, датчики руху, відеокамери та систему сповіщення. Це дозволяє відслідковувати рух автомобілів, виявляти незвичайну активність і сповіщати відповідні служби про потенційні загрози або порушення безпеки.

- Відеоспостереження: система забезпечує нагляд за парковочними майданчиками за допомогою відеокамер, що дозволяє виявляти незвичайну активність або порушення безпеки.

- Система контролю доступу: забезпечує авторизований доступ до парковочних майданчиків шляхом використання карток або інших ідентифікаторів.

- Автоматичне розпізнавання номерних знаків: дозволяє автоматично визначати номерні знаки автомобілів, що в'їжджають на парковку, і порівнювати їх з базою даних для перевірки авторизації.

- Система виявлення вторгнень: використовує датчики та аналіз поведінки, щоб виявляти незвичайні або підозрілі дії, такі як вандалізм або несанкціонований доступ.

- Система контролю доступу: забезпечує обмежений доступ до парковочних майданчиків тільки авторизованим користувачам.

- Датчики руху: сповіщають про рух автомобілів на парковці і допомагають відслідковувати рух автомобілів у реальному часі.

- Відеокамери: записують відео з парковочних майданчиків, що дозволяє відстежувати події і дії користувачів.

- Система сповіщення: надсилає сповіщення адміністраторам або відповідним службам про потенційні загрози або порушення безпеки.

Аналізовані програмні рішення, такі як "Smart Parking Security Suite" і "Parking Security Management System", використовуються для забезпечення безпеки в системах розумного паркування. Вони пропонують широкий спектр функціональних можливостей, включаючи відеоспостереження, систему контролю доступу, автоматичне розпізнавання номерних знаків та систему

виявлення вторгнень. Ці рішення допомагають виявляти незаконну діяльність, забезпечують контроль доступу та сповіщають про потенційні загрози безпеці паркувальної системи.

Проте, при виборі програмного рішення для забезпечення безпеки в системі розумного паркування, важливо враховувати специфічні потреби і вимоги конкретної системи, а також забезпечити його інтеграцію з існуючими компонентами та інфраструктурою. Крім того, важливо постійно оновлювати програмне забезпечення та вживати заходів для запобігання новим загрозам безпеці.

Аналіз програмних рішень для забезпечення безпеки в системі розумного паркування є важливим кроком у розробці і вдосконаленні безпечних та надійних паркувальних систем, що сприяють покращенню зручності та безпеки для користувачів.

На ринку мобільних застосунків наразі доступні сотні додатків для паркування. Однак у цьому розділі вибираються додатки для паркування, які надають статистику зайнятості паркувальних місць у реальному часі та навігаційні напрямки до забронюваного місця паркування. Всі вищезгадані додатки підтримують бронювання, яке доступне тільки на закритих парковках. На закритій автостоянці програмне забезпечення для розумного паркування також дає навігаційні вказівки до забронюваного місця паркування. Розумні інструменти паркування складаються з датчиків, технологій і програм, які використовуються для виявлення людей на парковці та підвищення ефективності паркування. Нижче наведено декілька датчиків, які допомагають виявляти інформацію про зайнятість автостоянки. Датчики є типовим інструментом, який детально досліджувався в попередніх публікаціях.

Пасивний інфрачервоний датчик: ці датчики виявляють зміни енергії, і коли автомобіль займає місце для паркування, ці датчики розпізнають зсув енергії та визначають зайнятість. Коли транспортний засіб або людина стоїть над датчиком, датчик виявляє зміну енергії. Його можна використовувати для виділення викидів

на основі кількості зміни енергії. Ці датчики, однак, чутливі до навколишнього середовища і будуть неточними за наявності снігу чи дощу. Пасивні інфрачервоні датчики повинні бути встановлені під землею або на даху. Як наслідок, вони вимагають значних витрат на придбання та обслуговування цих датчиків. Ці датчики підходять для критих закритих автостоянок, але не для відкритих відкритих автостоянок. Принцип роботи інфрачервоного датчика наведено на рисунку 1.4.

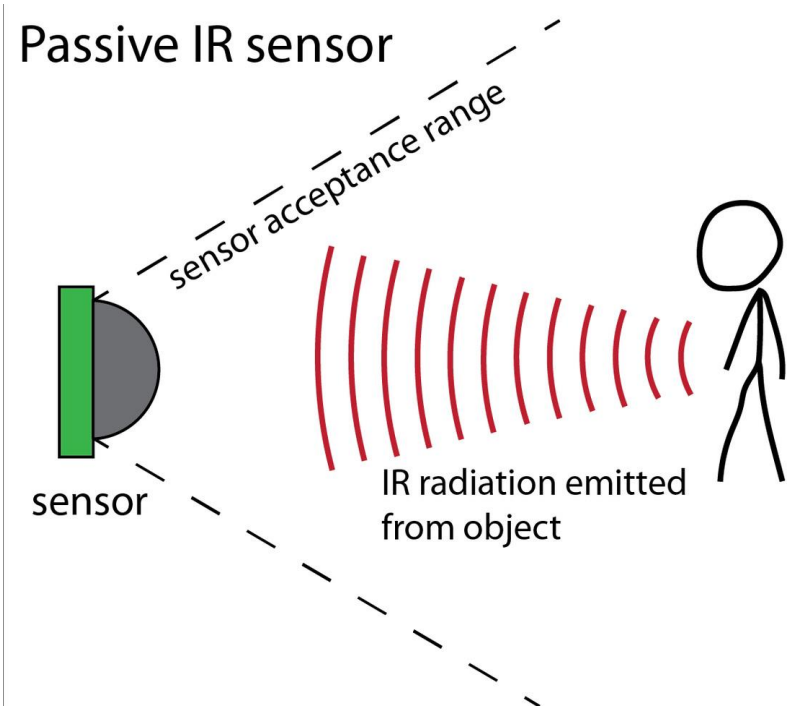


Рисунок 1.4 – Принцип роботи пасивного інфрачервоного датчика

Активний інфрачервоний датчик: ці датчики випромінюють інфрачервону енергію та використовують кількість відбитої енергії для виявлення будь-якого предмета чи автомобіля. Вони також чутливі до змін навколишнього середовища, таких як дощ або сніг. Як наслідок, вони потребують встановлення на всіх паркувальних місцях і потребують значних інвестицій та обслуговування. Встановлення датчиків на всіх паркувальних місцях допоможе отримати статус зайнятості паркінгу. Ці датчики часто встановлюються над головою і підходять для внутрішніх закритих автостоянок. Оскільки ці датчики чутливі до змін

навколишнього середовища, вони не підходять для відкритих автостоянок. Принцип роботи активного інфрачервоного датчика наведений на рисунку 1.5.

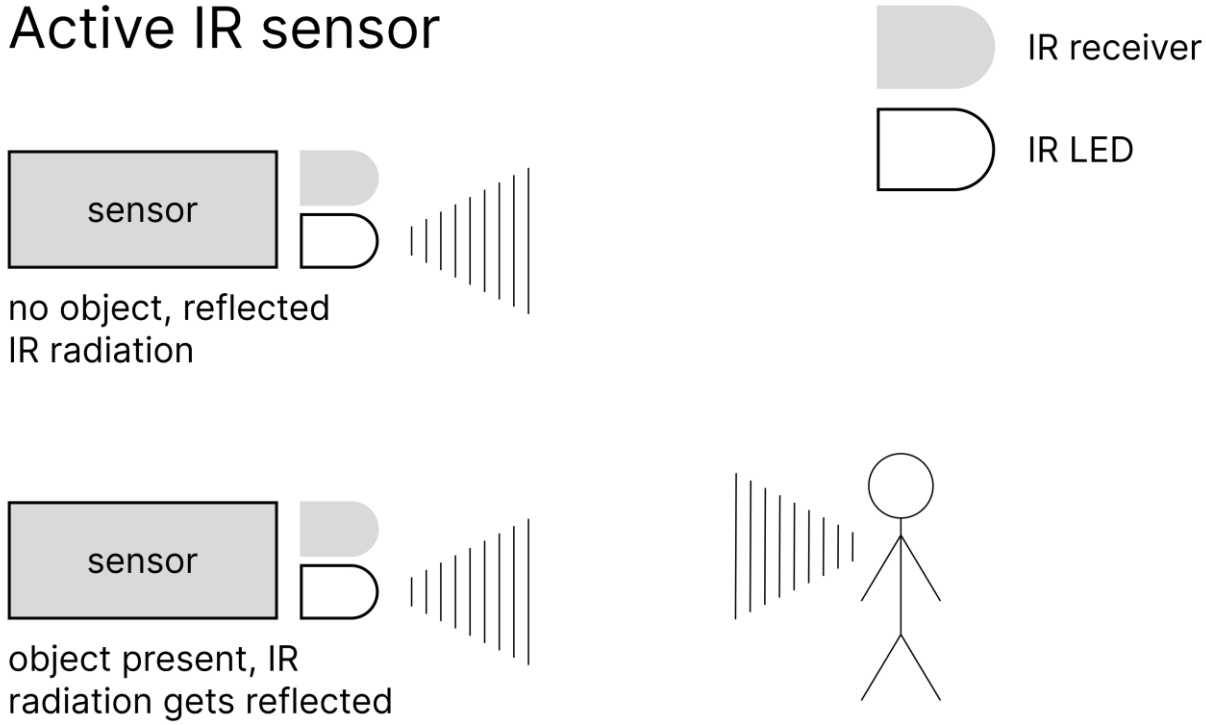


Рисунок 1.5 – Принцип роботи активного інфрачервоного датчика

Ультразвукові датчики: генерують звукові хвилі в діапазоні від 25 до 50 кГц і ідентифікував об’єкти на основі відбитої енергії. Вони часто встановлюються на стелі і чутливі до змін навколишнього середовища, таких як дощ і сніг. Як наслідок, вони більше підходять для закритих парковок, ніж відкритих парковок. Він може відрізнити транспортний засіб від людини на основі відстані, на якій відбиваються хвилі. Ці датчики слід розташувати вгорі кожного паркувального місця, щоб отримати статус зайнятості паркування. Ці датчики були б недорогими, але встановлення та технічне обслуговування кількох датчиків, а також підключення їх до мережі було б дорогим у довгостроковій перспективі. Бездротові ультразвукові датчики також використовуються для збору інформації про зайнятість автостоянки. Бездротові сенсорні мережі, такі як протокол ZigBee або інші подібні мережі, з’єднують їх. З іншого боку, бездротові датчики

потребують регулярного обслуговування. Інше дослідження використовує ультразвукові датчики на автомобілі, що проїжджає, і дані про зайнятість паркувальних місць збираються на регулярній основі. Проїжджаючий автомобіль не може отримати інформацію про заповненість автостоянки в реальному часі.

Індуктивні петлеві детектори встановлюються за допомогою підземної системи електропроводки та використовують електромагнітні принципи для виявлення присутності автомобіля. Зазвичай вони використовуються на в'їзді та виїзді, щоб визначити кількість присутніх автомобілів, що може бути використано для визначення наявності місць для паркування. Ці детектори дорогі в установці та обслуговуванні, але вони зазвичай використовуються на критих автостоянках для підрахунку доступних паркувальних місць. Ці детектори, які використовуються на кількох комерційних стоянках, пропонують точний підрахунок автомобілів на закритій стоянці. Проте статус зайнятості окремої стоянки не можна визначити за допомогою індуктивних петлевих детекторів.

1.4 Висновки

Отже, у першому розділі було проведено огляд існуючих методів та способіврealізації кіберфізичної системи для розумного паркування. Серед них: спосіб за допомогою ультрамагнітних давачів, інфрачервоних давачів, технології блокчейн та індуктивних петлевих детекторів.

Також було проаналізовано переваги та недоліки існуючих методів та засобів та дійдено висновку, що зазначені вище способи є дороговартісними та складними у встановленні, монтажі та обслуговуванні, а також складно масштабуються. Тому для подальшої роботи було обрано спосіб реалізації кіберфізичної системи для розумного паркування за допомогою камер зовнішнього спостереження. Цей спосіб є економічнішим та помітно легше масштабується та легший у монтажі та обслуговуванні.

2 ПРОБЛЕМА БЕЗПЕКИ КІБЕРФІЗИЧНОЇ СИСТЕИМИ ДЛЯ РОЗУМНОГО ПАРКУВАННЯ

2.1 Клієнт-серверна архітектура та можливі фактори загрози безпеці кіберфізичної системи

Оскільки пропонована у роботі кіберфізична система для розумного паркування, яка зображена на рисунку 2.1, складається з двох частин – апаратної частини (камери та всі фізичні пристрої, необхідні для функціонування) та програмної частини (клієнтська та серверна підсистеми), необхідно дослідити можливі фактори, які можуть вплинути на безпеку цієї системи.

Рисунок 1.5 – Принцип роботи активного інфрачервоного датчика

Якщо апаратне забезпечення можна просто перевірити на надійність і продуктивність, програмна підсистема потребує глибшого дослідження. Враховуючи те, що система має як серверну, так і клієнтську частину, було проведено аналіз факторів, які впливають на безпеку обох частин цієї кіберфізичної системи. Серед них: неправильна конфігурація безпеки, ін’єкції на

стороні клієнта (незахищені дані автентифікації, зловмисне програмне забезпечення), недостатній захист транспортного рівня (атаки MITM), незахищене зберігання даних (база даних), рутування пристрою/джейлбрейк, зворотне проектування, розкриття конфіденційних даних (порушення приватних даних) , неналежне реєстрування та моніторинг. Результати аналізу представлені в схематичному вигляді на рисунку 2.2.

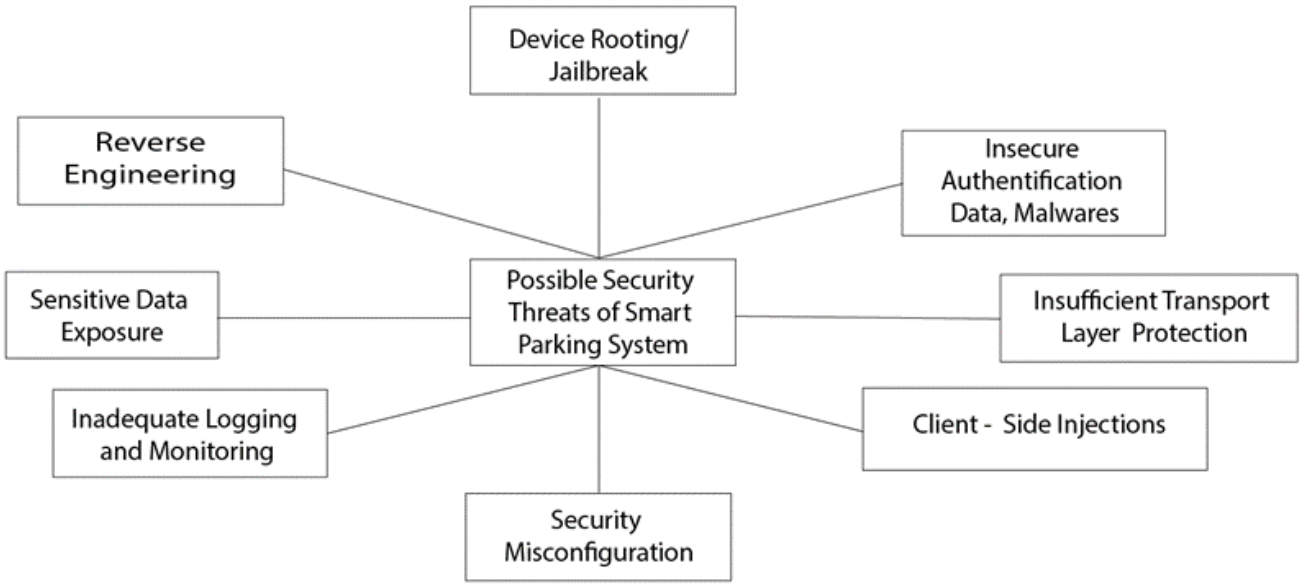


Рисунок 2.2 - Можливі загрози безпеці кіберфізичної системи для розумного паркування

2.1.1 Ризики безпеки на стороні клієнта

Оскільки клієнтська частина має бути розроблена у вигляді кросплатформного мобільного додатку, який не передбачає зберігання будь-якої приватної інформації, такої як особистий номер телефону, логін і пароль, може здатися, що вона менш сприйнятлива до атак з боку хакерів або витоку інформації. Однак можливість ризиків, пов'язаних із безпекою системи на стороні клієнта, не може бути повністю виключена. Серверна частина програмного забезпечення, навпаки, дуже чутлива, оскільки містить алгоритми розпізнавання зображень автомобіля за допомогою штучної нейронної мережі.

Несанкціонований доступ до бази даних, програмного коду або системних файлів може призвести до некоректної роботи алгоритмів і, як наслідок, до надання клієнтській частині некоректної інформації про зайнятість або незайнятість паркувального місця. Тобто некоректна робота всієї системи в цілому. Крім того, оскільки зв'язок між клієнтською та серверною частинами має бути реалізований за допомогою інтерфейсу прикладного програмування (API), додаткові вузькі місця з'являються в системі безпеки розумного паркування.

Для більшої зручності користувача та швидшого доступу до системи було вирішено розробити клієнтську частину у вигляді кросплатформного мобільного додатку. За останнє десятиліття індустрія розробки мобільних додатків значно зросла, але й кіберзлочини не залишилися на попередньому етапі. Все це призвело до того, що неможливо завантажити мобільний додаток у Google Play Store або Apple App Store без перевірки показників безпеки та впевненості, що додаток не буде звинувачено у витоку інформації чи шахрайстві з особистими даними. Але безпека мобільних програм — це більше, ніж просто захист їх від шкідливого програмного забезпечення та зовнішніх загроз. Спочатку нам потрібно визначити основні принципи безпеки відкритих веб-додатків та їх основні загрози безпеці, щоб мати можливість аналізувати заходи безпеки та розробляти методи та інструменти для підвищення рівня їх безпеки [3].

Неналежне використання функцій смартфона або непередбачувані збої під час використання налаштувань контролю безпеки. Це стосується налаштувань конфіденційності, дозволів, неправильного використання Touch ID, FaceID, Keychain тощо.

Достатнім вузьким місцем, яке часто можна зустріти під час вирішення проблем безпеки мобільних додатків, є відсутність безпечної системи зберігання даних. Розробники мобільних пристроїв зазвичай покладаються на сховище клієнтського пристрою для деяких особистих і внутрішніх даних. Але якщо хакери отримують доступ до пристрою або пристрій може бути вкрадено чи втрачено, ці дані можуть бути використані для зловмисних цілей. В результаті це

призводить до таких кіберзлочинів, як порушення політики конфіденційності та викрадення персональних даних з метою їх зловмисного використання.

Під час розробки мобільних додатків передача даних відбувається за моделлю клієнт-сервер. Таким чином, коли дані передаються, вони можуть бути перехоплені зловмисниками через Інтернет. Зловмисники також можуть перехоплювати дані під час банківського переказу. Передача даних через ненадійні канали зв'язку призводить до порушення політики конфіденційності, крадіжки персональних даних, шахрайства та втрати ділової репутації компанії.

Зловмисники або боти можуть отримати дані під час автентифікації та проникнути в обліковий запис користувача. Це може призвести до витоку особистої інформації, крадіжки персональних даних і несанкціонованого доступу до внутрішніх даних облікового запису користувача.

Зловмисники або рекламні боти можуть мати доступ до даних, які не були зашифровані або захищені належним чином. Це може призвести до несанкціонованого доступу до внутрішніх даних програми, крадіжки даних, витоку особистої інформації користувачів тощо.

Зловмисники можуть перехоплювати дані під час процесу авторизації та використовувати їх для несанкціонованого доступу до програми. Як наслідок, це призводить до витоку особистої інформації та втрати ділової репутації компанії.

Низька якість коду може призвести до непередбачуваних збоїв програми або виникнення численних помилок під час використання. Крім того, це знижує продуктивність програми та може призвести до надмірного використання пам'яті або повільного завантаження графічних елементів в інтерфейсі користувача під час роботи.

Зловмисники, отримуючи доступ до вихідного коду, можуть інтегрувати в нього рекламу чи шкідливі скрипти або замінювати частини коду, що може призвести до некоректної роботи програми, втрати деяких функцій або заміни певного функціоналу для використання програми. для зловмисних цілей.

Ризики, які впливають на безпеку мобільних застосунків представлені у вигляді схеми на рисунку 2.3.

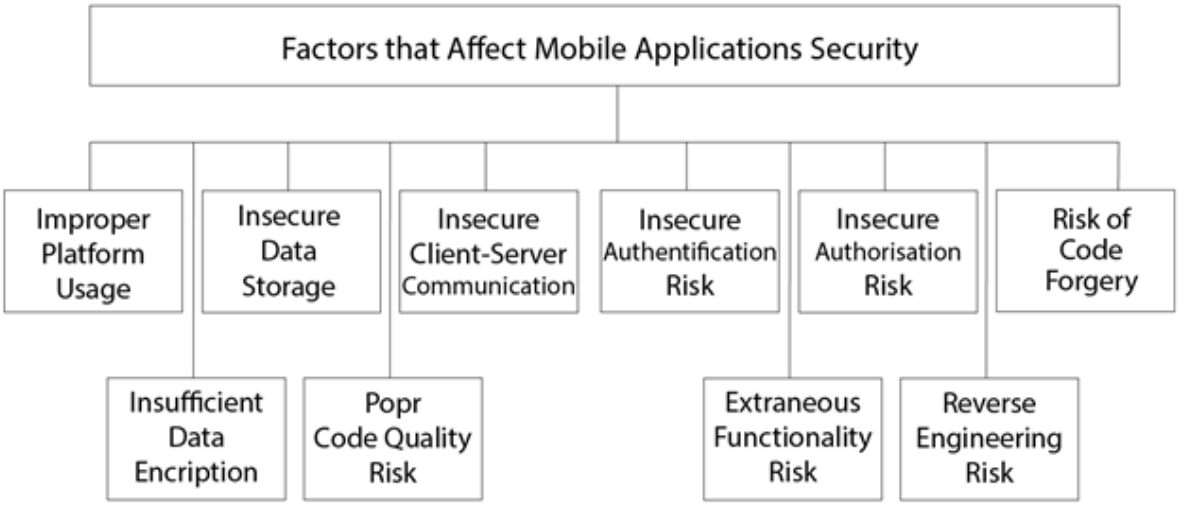


Рисунок 2.3 – Ризики, які впливають на безпеку мобільних застосунків

Також було зібрано статистичні дані щодо частоти виникнення кожного фактору ризику щоб визначити фактори, які зустрічаються найчастіше а значить, відповідно, є найнебезпечнішими для клієнтської частини кіберфізичної системи для розумного паркування.

Частота прояву факторів, що впливають на безпеку мобільних додатків, зображена у вигляді стовпчикової діаграми на рисунку 2.4.

Відповідно до статистичних даних, наведених на рисунку 2.4, незахищене зберігання даних і незахищений зв’язок клієнт-сервер є найчастішими причинами ризиків безпеки мобільних застосунків.

Зловмисники можуть завантажити мобільний додаток, щоб переробити його функції. Тобто одна і та ж програма в різних версіях може працювати абсолютно по-різному.

У цьому випадку зловмисники перевіряють функції мобільного додатку, щоб знайти вузькі місця та запровадити сторонній код.

Також зловмисники можуть впроваджувати шкідливі зміни в мобільний додаток, що дозволяє їм змінювати його функціонал.

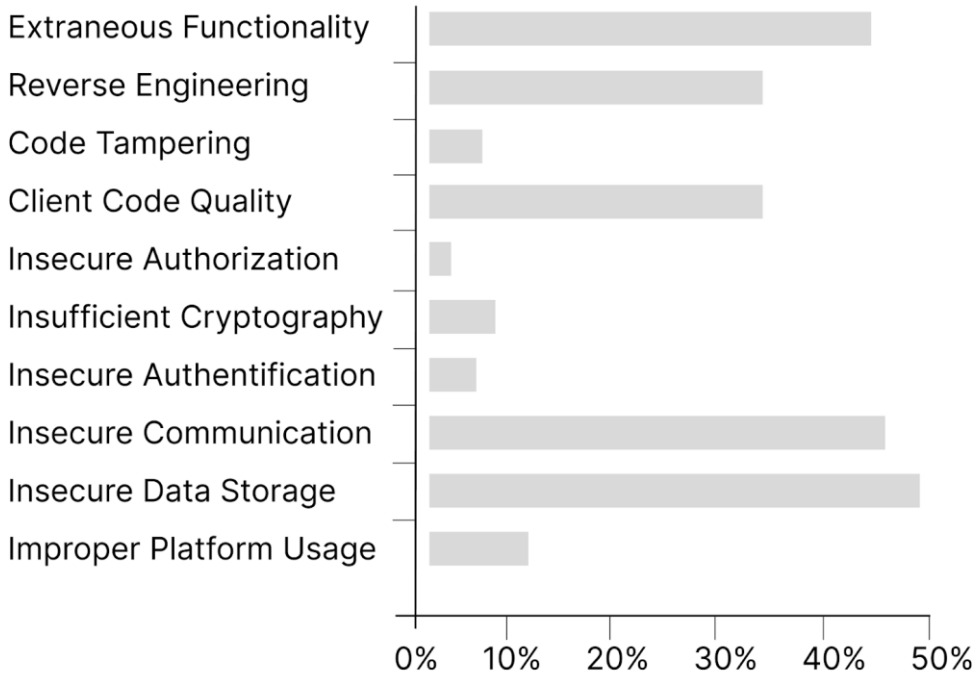


Рисунок 2.4 – Частота прояву факторів, що впливають на безпеку мобільних застосунків

2.1.2 Ризики безпеки на стороні сервера

Одним з найважливіших факторів забезпечення безпеки даних на стороні сервера є важливість шифрування.

Обмін миттєвими повідомленнями є одним із найпоширеніших способів спілкування клієнта з сервером на рівні клієнт-серверної архітектури.

Таким чином, зашифровані текстові повідомлення є важливим заходом безпеки.

Тому розробники мобільних застосунків повинні звернути увагу на цей захід безпеки. Однак інформаційна безпека - це не єдина перевага, яку можна отримати від полегшення зашифрованих текстових повідомлень. Фактично, є також деякі інші способи забезпечення безпеки даних на стороні сервера, які наведено у таблиці 2.1.

Таблиця 2.1 – Способи забезпечення безпеки даних на стороні сервера

Спосіб	Опис
Безпечна передача даних	Не можна заперечувати, що зашифровані дані забезпечують безпеку під час передачі. Якщо користувачі надсилають файли електронною поштою або поширюють їх через хмарний сервер, вони можуть використовувати шифрування. Це допомагає їм переконатися, що жоден неавторизований користувач не зможе переглянути цю інформацію.
Забезпечення цілісності даних	Неправомірне використання даних може відбуватися не лише шляхом цілеспрямованої крадіжки даних, але й шляхом маніпулювання. Хакер може маніпулювати конкретними даними, щоб порушити корпоративні комунікації. Однак використання зашифрованих даних може уникнути такої ситуації.
Захист даних на всіх пристроях	Різноманітність пристроїв стає значною частиною нашого життя. Однак передача даних з одного пристрою на інший – справа ризикована. Отже, технологія шифрування може захищати та зберігати дані на всіх пристроях. Такий захід безпеки може стримувати неавторизованих користувачів.
Шифрування даних	У більшості випадків видно, що багато галузей потребують суворого дотримання вимог щодо захисту всіх тих, чия особиста інформація зберігається організаціями. FIPS, HIPAA чи будь-які інші нормативні акти повністю покладаються на шифрування для захисту даних

2.2 Ризики безпеки API

Application Programming Interface (API) – це тип програмного забезпечення, яке підключається до функціональних можливостей програми та економить час розробників. Часто за допомогою API підключається функціональність, створена іншими розробниками, або часто використовувана функціональність, або виконується підключення клієнт-сервер. Це допомагає розробникам економити час і не розробляти з нуля функції, які вже є загальнодоступними. Коли справа доходить до з'єднання кількох частин програмної системи разом, це дійсно корисно [17]. Однак під час використання API існують також ризики для безпеки. Існує дві основні причини, чому під час використання API слід враховувати питання безпеки.

3) Простий спосіб отримати доступ до внутрішньої інформації програми – через API можна отримати доступ до збережених даних, включаючи особисту інформацію користувача (логін, пароль тощо) з метою несанкціонованого розповсюдження або зловмисної діяльності.

4) Простий спосіб для зловмисників обійти заходи безпеки, навіть якщо брандмауер увімкнено. Тому не варто нехтувати добре продуманою стратегією безпеки.

Існує значна різниця в заходах безпеки для традиційних веб-програм і веб-програм на основі API. Ця різниця полягає в їхній архітектурі та тому, як вони побудовані. Раніше захист веб-додатків вимагав лише захисту портів HTTP і HTTPS.

Сучасні програми, які використовують кілька API і різні протоколи, потребують комплексного захисту всіх частин програми, враховуючи всі її вузькі місця. Це особливо важливо, коли API розширює свою функціональність, що ускладнює керування безпекою. Крім того, під час заміни API попередньо розроблені заходи безпеки необхідно переглянути та переналаштувати вручну. Різниця в структурі API-додатків робить їх чутливими до зовнішніх загроз [4]:

API зазвичай захищені за допомогою веб-токена JSON або ключа API. Це дозволяє захистити API і, у разі виявлення незвичної або підозрілої поведінки, закрити доступ до ключів API.

Захист від DDoS-атак в основному побудований на принципі відхилення запитів від підозрілих акторів. Це стає складніше, оскільки в додатках на основі API кожен трафік виглядає підозрілим.

Сервер відповідає за зв'язок між програмою та користувачем за екраном мобільного телефону. Основною причиною вразливості сервера є те, що інколи розробники недостатньо серйозно ставляться до належних заходів безпеки та захисту підключень до сервера під час роботи з API.

Відповідно до досліджень у всьому світі порушення та витік даних зазвичай відбуваються, коли відбувається недостатнє реєстрування.

На відміну від автентифікації, процес авторизації в кожній програмі має власну логіку, і це часто може бути вузьким місцем для зловмисників. Якщо процес авторизації недостатньо продуманий і захищений, хакери можуть увійти в систему і отримати доступ до даних за допомогою ітеративного методу вибору ідентифікатора [4].

Згідно зі статистичними даними, основними та найпоширенішими факторами є порушення даних, випадкове розкриття даних, застарілі API, відмова в обслуговуванні, невідомі або тіньові API та захоплення облікового запису. Частота прояву вищезазначених факторів, які впливають на безпеку прикладного програмного інтерфейсу (API), представлена на рисунку 2.5.

Тобто проміжне програмне забезпечення забезпечує додатковий захист сервера від підозрілих або зловмисних запитів шляхом їх перехоплення та перевірки. І тільки якщо запит безпечний, він відправляється на сервер для подальшої обробки.

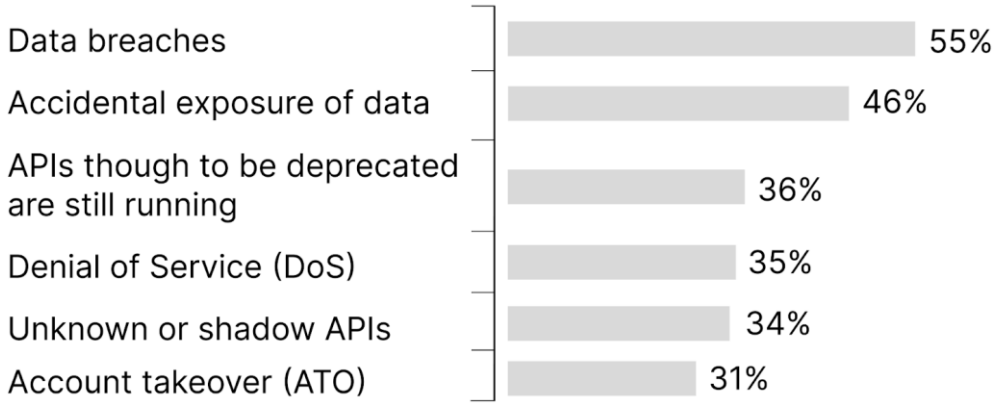


Рисунок 2.5 – Частота прояву факторів, що впливають на безпеку прикладного програмного інтерфейсу (API)

2.3 Вибір методів і середовища реалізації програмного забезпечення

Розглядаючи фактори, які впливають на окремі частини програмної системи розумного паркування, було вирішено взяти до уваги ті, які найчастіше впливають на безпеку системи та залучають її різні частини з різних точок зору (наприклад, доступ до даних, зв’язок клієнт-сервер). , вузькі місця при використанні API) і прийти до рішення, яке допоможе врахувати їх у комплексі.

Пропоноване рішення є проміжним програмним забезпеченням для додаткової перевірки запитів від клієнта до сервера. Це ефективний інструмент для виконання операцій або обчислень всередині з’єднання «запит-відповідь» у моделі взаємодії клієнт-сервер. Його слід використовувати, коли необхідно виконати певну операцію або перевірити надійність запиту не безпосередньо на сервері з міркувань безпеки.

Тому було вдосконалено архітектуру кіберфізичної ситсеми для розумного паркування, наведену на рисунку 2.1, додавши проміжне програмне забезпечення безпеки до серверної частини програмного забезпечення. Пропонована архітектура кіберфізичної системи для розумного паркування, із застосуванням проміжного програмного забезпечення, представлена на рисунку 2.6. З такою

архітектурою набагато простіше визначити, чи справді запит був надісланий з нашого клієнтського додатку, і перевірити, чи він не шкідливий і чи не містить підозрілого коду. Також це проміжне програмне забезпечення скоротить час роботи програми, якщо запит не є відповідним, оскільки його результат вже відомий, оскільки виклик Google Cloud API не відбувається миттєво.

Розглянемо приклад роботи запропонованої програмної архітектури розумного паркування з використанням Security Middleware. Для кращого розуміння та наочності розглянемо алгоритм, запропонований на рисунку 2.6.

Згідно з рисунком 2.6 запит надсилається на сервер із мобільного клієнтського додатку. Однак запит не завжди може бути безпечним. Щоб перевірити це, ми перевіряємо запит за допомогою інтегрованого проміжного програмного забезпечення.

Це дозволить нам переконатися, що запит дійсно надійшов з нашого клієнтського додатку, а не з якогось стороннього ресурсу, і що запит безпечний. Також використання проміжного програмного забезпечення скоротить час роботи програми, якщо запит є невідповідним або його результат вже відомий.

Оскільки виклик Google Cloud Vision API не є миттєвим, ми можемо використати цей час для перевірки безпеки отриманого запиту.

Якщо запит безпечний, він відправляється на подальшу обробку, а саме на перевірку вибраного користувачем паркувального місця на зайнятість. Якщо запит визначається як потенційно шкідливий або сторонній, він нейтралізується в першому випадку, а відповідь сервера на такий запит ігнорується.

Для розробки проміжного програмного забезпечення (security middleware) для підвищення безпеки кіберфізичної системи для розумної парковки можна використати різні технології в залежності від конкретних вимог і обмежень розроблюваної системи.

Перелік технологій, які можливо використати для розробки проміжного ПЗ для забезпечення безпеки кіберфізичної системи для розумної парковки наведено у таблиці 2.2.

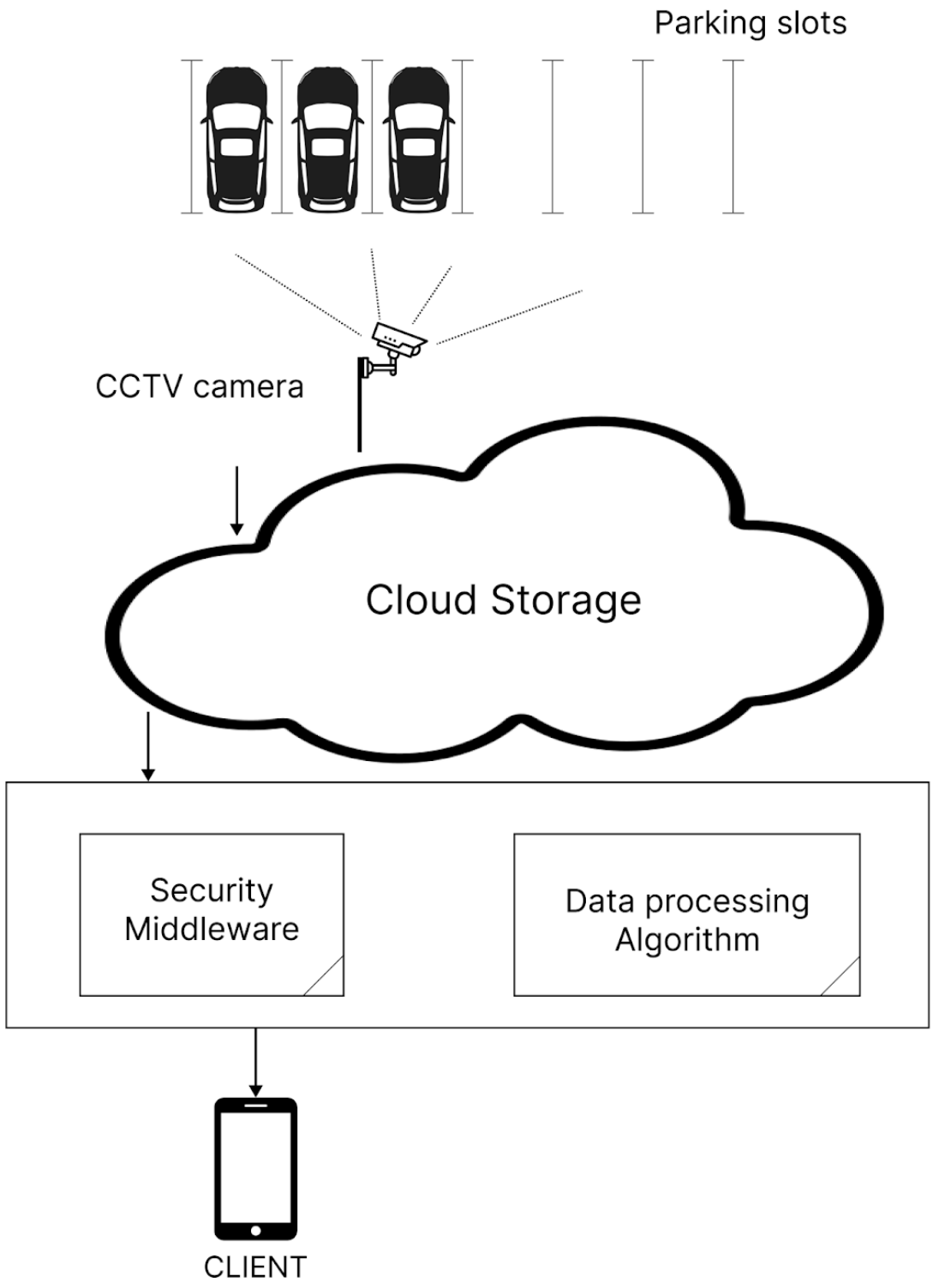


Рисунок 2.6 – Архітектура кіберфізичної системи для розумного паркування, із застосуванням проміжного програмного забезпечення

Таблиця 2.2 – Перелік технологій для розробки проміжного ПЗ для забезпечення безпеки кіберфізичної системи для розумної парковки

Вид технології	Короткий опис
Мова програмування	Можна використовувати мову програмування, зручну і відповідну для вашого використання. Наприклад, мови програмування, які часто використовуються для розробки проміжного програмного забезпечення, включають Java, C++, Python та Node.js.
Фреймворки:	Використання фреймворків може значно полегшити розробку проміжного програмного забезпечення. Наприклад, для веб-програмування можна використовувати фреймворки, такі як Django (Python), Spring (Java), Express.js (Node.js) або ASP.NET (C#). Для розробки мікросервісів можна використовувати фреймворки, такі як Spring Boot (Java), Flask (Python) або Nest.js (Node.js).
Бази даних	Для збереження даних можна використовувати реляційні бази даних, наприклад, MySQL або PostgreSQL, або нереляційні бази даних, такі як MongoDB або Cassandra, в залежності від потреб розроблюваної системи.
Комунікація та протоколи	Для забезпечення безпеки комунікації можна використовувати протоколи шифрування, такі як HTTPS, TLS або SSH. Для взаємодії зі зовнішніми сервісами або іншими компонентами системи можна використовувати веб-сервіси, такі як REST або GraphQL, або інші протоколи, такі як MQTT для IoT-пристроїв.

Кінець таблиці 2.2 – Перелік технологій для розробки проміжного ПЗ для забезпечення безпеки кіберфізичної системи для розумної парковки

Аутентифікація та авторизація:	Для забезпечення безпеки доступу до системи ви можете використовувати механізми аутентифікації та авторизації, такі як JWT (JSON Web Tokens), OAuth або OpenID Connect.
Моніторинг та логування	Для відстеження роботи системи та виявлення можливих проблем можна використовувати інструменти моніторингу, такі як Prometheus або ELK Stack (Elasticsearch, Logstash, Kibana), які допоможуть вам аналізувати логи та метрики вашої системи.
Тестування	Не забувайте про важливість тестування вашого проміжного програмного забезпечення. Ви можете використовувати фреймворки для автоматизованого тестування, такі як JUnit (Java), pytest (Python) або Mocha (Node.js), або інші інструменти для юніт-тестування, інтеграційного тестування та навантажувального тестування.

Рекомендовано провести детальний аналіз потреб кіберфізичної системи та проконсультуватися з експертами з безпеки, щоб вибрати найкращий стек технологій для пропонованого проміжного програмного забезпечення.

2.4 Висновки

Отже, у даному розділі було проведено аналіз потенційних загроз, які впливають на безпеку кіберфізичної системи для розумного паркування. Оскільки пропонована кіберфізична система базується на клієнт-серверній архітектурі, визначено потенційні загрози клієнтській частині та серверній частині

програмного забезпечення. Також, оскільки клієнтська частина буде приєднана до серверної частини за допомогою прикладного програмного забезпечення (API), було розглянуто фактори, які впливають на безпеку API.

Також у ході даного розділу було визначено стек технологій, який може бути використаний для розробки проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи “Розумна парковка”.

3 ВПРОВАДЖЕННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ ДЛЯ ПІДВИЩЕННЯ БЕЗПЕКИ КІБЕРФІЗИЧНОЇ СИСТЕМИ “РОЗУМНА ПАРКОВКА”

3.1 Запропонований спосіб підвищення безпеки кіберфізичної системи “Розумна Парковка”

Оскільки в попередньому розділі було розглянуто стек технологій, який дає можливість розробити проміжне програмне забезпечення для підвищення безпеки кіберфізичної системи “Розумна парковка”, серед запропонованих технологій було обрано фреймворк ASP.NET Core. Однією з найважливіших особливостей ASP.NET Core є структура «Middleware». Це дуже ефективна функція для виконання індивідуальних операцій у середині моделі «Запит-відповідь» і керування трафіком запит-відповідь. Ми можемо виконувати багато різних завдань, наприклад перевіряти дійсність вхідного запиту, створювати відповіді з кешу в ASP.NET Core за допомогою структур, які збираються разом.

Структура, яка дозволяє виконувати методи, додані як доповнення до класу, отриманого з інтерфейсу IApplicationBuilder на основі проміжного програмного забезпечення. Ми використовуємо його, коли хочемо виконати різні операції та дати інший напрямок ходу процесу, поки відповідь на запит не надійде від клієнта до веб-додатку.

Middleware — це термінологічна структура, яка описує проміжне програмне забезпечення. Це структура, яка всюди має однаковий механізм роботи. Проміжне програмне забезпечення запускається повільно. Коли проміжне програмне забезпечення запускається, воно запускає інше проміжне програмне забезпечення до закінчення терміну його дії. Схема роботи компонентів проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи “Розумна парковка” наведено на рисунку 3.1.

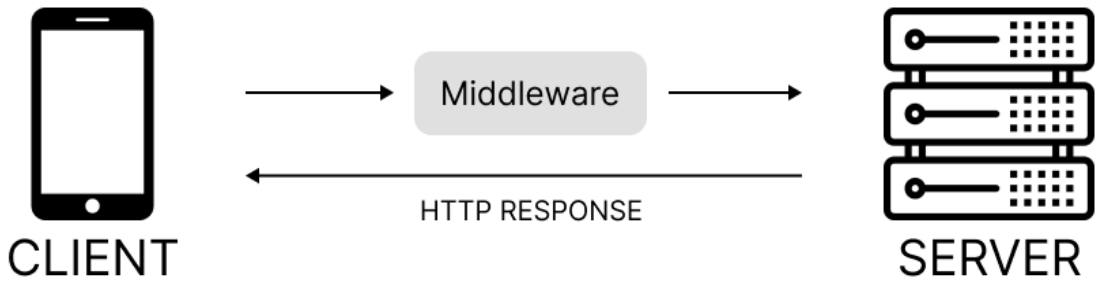


Рисунок 3.1 – Схема роботи компонентів кіберфізичної системи “Розумна парковка” з урахуванням проміжного програмного забезпечення

Принцип роботи проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи “Розумна парковка” з клієнтсьою частиною кіберфізичної системи, яка пропонується у формі кросплатформного мобільного застосунку, полягає в надходженні запитів від клієнта до проміжного програмного забезпечення та наданні відповіді з уже перевіреними та безпечними даними. Якщо дані з певних причин не проходять перевірку за допомогою проміжного програмного забезпечення, відповідь на такий запит не надається, а дані самого потенційно шкідливого запиту знищуються проміжним програмним забезпеченням. Принцип роботи проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи “Розумна парковка” представлено на рисунку 3.2.

На схемі, представленій на рисунку 3.2 отримано запит і запущено одне проміжне програмне забезпечення, і в цьому місці виконуються запитані операції, а потім наступна команда запускає наступне проміжне програмне забезпечення. Він продовжує запускатися таким чином до третього проміжного програмного забезпечення, потім, оскільки немає проміжного програмного забезпечення для запуску, воно повертається до попереднього проміжного програмного забезпечення, завершує його та повертається до попереднього, а коли останнє завершено, відповідь повертається. Таким чином, спіраль завершена.

Asp.Net Core має ядро, яке структурно підтримує структурування проміжного програмного забезпечення. Усі функції в методі Configure у файлі запуску фактично діють як проміжне програмне забезпечення. У структуруванні Asp.Net Core проміжне програмне забезпечення починається з назви «Використовувати» та викликається в налаштуваннях.

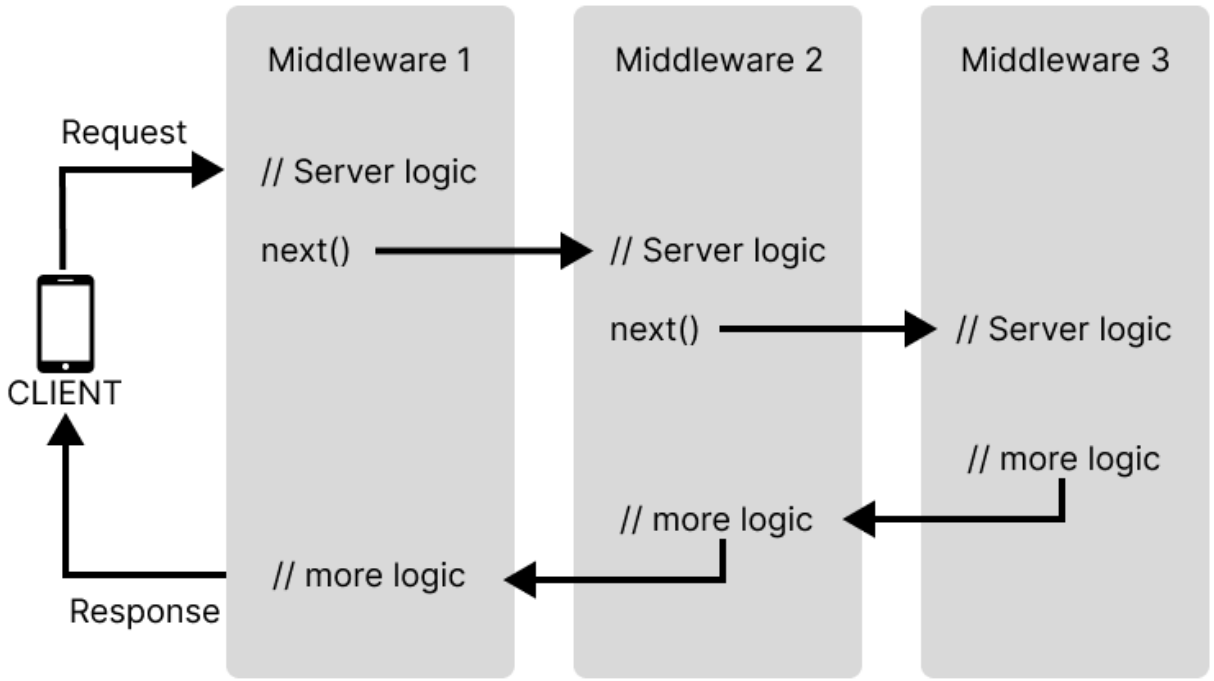


Рисунок 3.2 – Принцип роботи проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи “Розумна парковка”

Порядок тригерів важливий у Middlewares. Тому проміжне програмне забезпечення слід сортувати, звертаючи увагу на робочий пріоритет. Наприклад, якщо потрібно виконати автентифікацію та авторизацію, спочатку слід викликати проміжне програмне забезпечення автентифікації, а потім — проміжне програмне забезпечення авторизації. Ігнорування порядку тут призведе до логічних помилок.

В Asp.Net Core є проміжне програмне забезпечення, встановлене в ядрі: Run, Use, Map, MapWhen. У таблиці 3.1 наведено опис класів та модулів проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи для розумної парковки на платформі ASP.NET Core.

Таблиця 3.1 – Опис класів та модулів проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи для розумної парковки.

Назва класу	Опис та призначення
SecurityMiddleware	Головний клас проміжного програмного забезпечення, який виконує перевірку безпеки перед обробкою запиту. Цей клас реалізує інтерфейс `IMiddleware` і містить метод `InvokeAsync`, в якому виконується логіка безпеки та виклик наступного кроку обробки.
AuthenticationService	Клас, відповідальний за аутентифікацію користувачів. Він містить методи для перевірки ідентифікаційних даних користувача, таких як логін та пароль, та повертає об'єкт `ClaimsPrincipal` з аутентифікованим користувачем.
AuthorizationService	Клас, відповідальний за авторизацію користувачів. Він містить методи для перевірки прав доступу користувача, наприклад, на основі ролей, дозволів або інших критеріїв. Він повертає логічне значення, що вказує на те, чи має користувач необхідні права доступу.
UserContext	Клас, який представляє контекст користувача. Він містить інформацію про аутентифікованого користувача,

	таку як ідентифікатор, ім'я, ролі тощо.

Кінець таблиці 3.1 – Опис класів та модулів проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи для розумної парковки.

Controllers	Модуль, який містить контролери, які обробляють запити користувачів. Контролери можуть використовувати `AuthenticationService` і `AuthorizationService` для перевірки безпеки та доступу до ресурсів.
Models	Модуль, який містить моделі даних, необхідні для аутентифікації та авторизації, наприклад, модель користувача або прав доступу.
Extensions	Модуль, який містить розширення для реєстрації сервісів та налаштування middleware в ASP.NET Core. Наприклад, можна створити метод розширення, який додає `SecurityMiddleware` до конвеєра обробки запитів.

Методи класів проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи “Розумна парковка” представлені у таблиці 3.2.

Таблиця 3.2 – Методи класів проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи для розумної парковки

Назва методу	Опис
Модуль запуску	Запускає проміжне програмне забезпечення. У результаті

	pupline не продовжується і дає прямий вихід. Цей ефект називається коротким замиканням. Його можна

Кінець таблиці 3.2 – Методи класів проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи для розумної парковки

	використовувати відповідно до операції, яка буде виконуватись.
Використання	Метод використання викликає наступне проміжне програмне забезпечення в процесі після його активації та має структуру, яка може повернутися назад і продовжити роботу після завершення нормальної функції проміжного програмного забезпечення.
Карта	Іноді нам може знадобитися відфільтрувати проміжне програмне забезпечення відповідно до шляху, який надсилає запит. Для цього ми можемо забезпечити контроль у функціях «Використовувати» або «Виконати» або виконувати більш професійні операції за допомогою методу «Карта».
MapWhen	За допомогою методу Map фільтрація виконується лише відповідно до шляху, за яким зроблено запит, фільтрація виконується відповідно до будь-якої функції вхідного запиту за допомогою методу MapWhen.

Для уніфікації та створення загального механізму перехоплення помилок за допомогою проміжного програмного забезпечення, яке постачається з .NET Core і реєстрації отриманих помилок було створено проект як WepApi та новий клас LoggingMiddleware у ньому. Він складається з конструктора та методу Invoke, який приймає простий параметр RequestDelegate. Ми перехоплюмо виняток,

відправивши запит у блокк try catch і запишемо помилку, яку було відстежено, у файл txt.

3.2 Структурна схема та алгоритм роботи програмно-технічного засобу підвищення безпеки кіберфізичної системи “Розумна Парковка”

При розробці архітектури кіберфізичної системи для розумного паркування було прийнято рішення розділити всю архітектуру системи умовно розділити на три рівні - зовнішній рівень, проміжний рівень та внутрішній рівень.

На зовнішньому рівні знаходиться паркувальна розмітка та камери зовнішнього спостереження, з яких відбуватиметься збір даних.

На проміжному рівні знаходиться обслуговуюче обладнання, а саме сервер, хмарна база даних та інтернет-з’єднання, за допомогою якого відбувається зв’язок між компонентами системи. до внутрішнього рівня відноситься програмне забезпечення, яке підтримує роботу системи та клієнт-серверну взаємодію.

Серед ключового програмного забезпечення було виділено проміжне програмне забезпечення для підвищення безпеки даної кіберфізичної системи та алгоритм обробки даних.

Архітектура програмного забезпечення кіберфізичної системи для розумної парковки з використанням проміжного програмного забезпечення для підвищення безпеки наведена на рисунку 3.3.

Архітектура програмного забезпечення для кіберфізичної системи розумної парковки з використанням проміжного програмного забезпечення для підвищення безпеки може бути організована за допомогою шарової або мікросервісної архітектури. Загальний опис архітектури кіберфізичної системи для розумної парковки, яка наведена на рисунку 3.3:

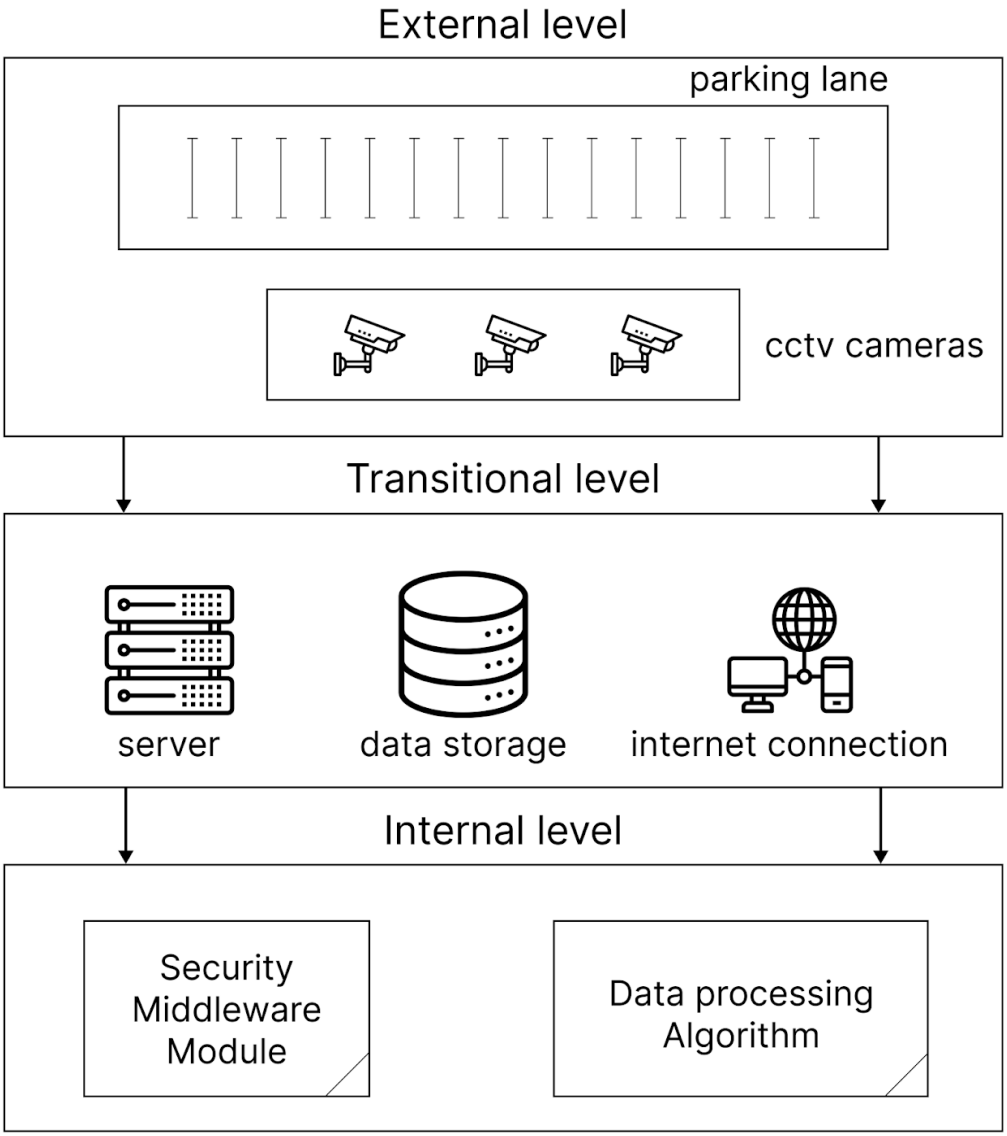


Рисунок 3.3 – Архітектура кіберфізичної системи для розумної парковки з використанням проміжного програмного забезпечення

1. Клієнтський інтерфейс включає в себе веб-додаток або мобільний додаток, яким користувачі можуть керувати розумною парковкою. Цей інтерфейс забезпечує взаємодію з користувачем, надсилає запити серверу та отримує відповіді.

2. Серверна сторона: включає проміжне програмне забезпечення, що забезпечує безпеку системи та зв'язок між різними компонентами. Проміжне програмне забезпечення безпеки - це серце системи, яке включає різні

компоненти безпеки, такі як аутентифікація, авторизація, шифрування, контроль доступу і логування.

Воно виконує перевірку безпеки і захисту системи від потенційних загроз.

Керуючий модуль відповідає за керування та координацію всіма компонентами системи розумної парковки.

Він приймає запити від клієнтського інтерфейсу і виконує необхідні дії, включаючи обробку операцій з парковкою, моніторинг стану системи і забезпечення безпеки. Компоненти парковки включають сенсори, бар'єри, камери тощо, які забезпечують збір даних про стан парковки та контроль доступу.

Вони взаємодіють з керуючим модулем для передачі даних та виконання відповідних дій.

База даних зберігає інформацію про користувачів, парковки, транзакції та інші важливі дані. Використовується для забезпечення постійного збереження та доступу до даних.

3. Безпека та захист. Цей компонент включає в себе всі заходи безпеки, необхідні для захисту системи. Він використовує проміжне програмне забезпечення безпеки для контролю доступу, аутентифікації, авторизації, шифрування та логування.

4. Моніторинг та аналіз. Цей компонент забезпечує моніторинг і аналіз стану системи, виявлення аномалій та подій безпеки. Він може включати системи журналювання, моніторингу мережі, виявлення вторгнень тощо.

Особливо важливо враховувати принципи безпеки, такі як захист даних, аутентифікація, авторизація та шифрування, і реалізувати їх відповідно до архітектури кіберфізичної системи для розумної парковки.

На рисунку 3.4 запропоновано алгоритм перевірки безпеки запиту від клієнта до сервера за допомогою прикладного програмного забезпечення.

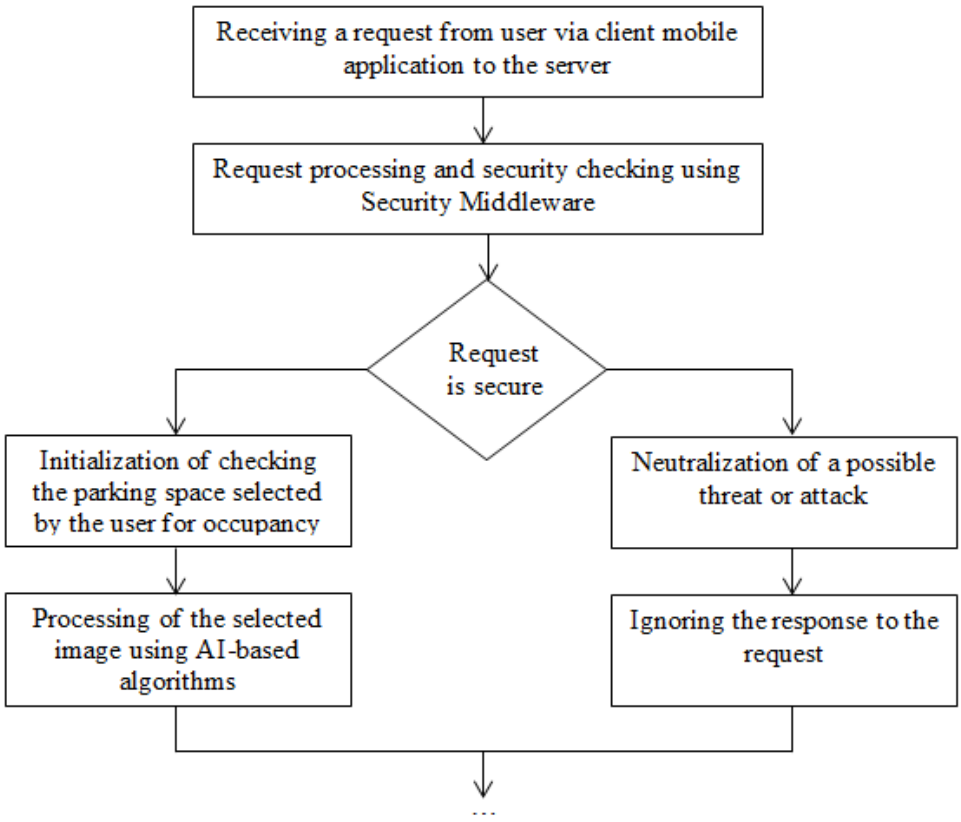


Рисунок 3.4 – Алгоритм перевірки безпеки запиту за допомогою проміжного програмного забезпечення

Алгоритм перевірки безпеки запиту від клієнта до сервера за допомогою проміжного програмного забезпечення у кіберфізичній системі для розумної парковки може включати наступні кроки:

1. Приймання запиту. Проміжне програмне забезпечення отримує запит від клієнта на сервер.

2. Аутентифікація. запит перевіряється на наявність правильних аутентифікаційних даних, таких як токен або ідентифікатор користувача. Це може включати перевірку логіну та паролю, або використання токенів доступу, таких як JWT (JSON Web Token).

3. Авторизація. Після аутентифікації запит перевіряється на доступ до ресурсу або виконання певної дії. Використовуються права доступу, ролі користувача або інші критерії, щоб визначити, чи має клієнт необхідні права для виконання запиту.

4. Шифрування. Якщо передаються конфіденційні дані, такі як паролі або особиста інформація, вони можуть бути зашифровані для забезпечення конфіденційності під час передачі через мережу.

5. Контроль доступу. Відбувається перевірка дозволу на доступ до конкретного ресурсу або операції. Це може включати перевірку прав доступу до бази даних або інших внутрішніх ресурсів.

6. Логування. Інформація про запит та його результат може бути записана в лог-файл для наступного аналізу та моніторингу безпеки системи.

7. Відправка запиту до наступного рівня обробки. Якщо запит успішно пройшов всі перевірки безпеки, він передається до наступного рівня обробки, де виконується необхідна бізнес-логіка.

Також важливо використовувати надійні криптографічні алгоритми та виконувати регулярне оновлення проміжного програмного забезпечення для забезпечення безпеки системи.

3.3 Висновки

Отже, у цьому розділі було запропоновано метод підвищення безпеки кіберфізичної системи “Розумна парковка”. Метод полягає у додаванні проміжного програмного забезпечення для підвищення безпеки у серверну частину кіберфізичної системи для розумного паркування.

Також було розроблено схему роботи компонентів кіберфізичної системи “Розумна парковка” з урахуванням проміжного програмного забезпечення, запропоновано принцип роботи проміжного програмного забезпечення для підвищення безпеки кіберфізичної системи “Розумна парковка”. Архітектура кіберфізичної системи для розумної парковки з використанням проміжного програмного забезпечення була також представлена у даному розділі.

ВИСНОВКИ

На даний момент актуальним завданням є розробка методів і засобів підвищення безпеки програмної системи розумної парковки. Метою даної роботи було проведення аналізу вимог до безпеки програмного забезпечення системи розумної парковки з метою виявлення частин програмного забезпечення, які є найбільш вразливими до зовнішніх загроз, і розробки методів і інструментів для підвищення їх безпеки.

У дипломній роботі запропоновано метод підвищення безпеки програмної системи розумна парковка на основі інтеграції проміжного ПЗ в архітектуру програмного забезпечення системи розумної парковки. Запропонований метод враховує всі критерії безпеки програмного забезпечення системи розумної парковки, тобто параметри безпечного доступу до бази даних, безпеки клієнтської програми, безпеки сервера та API безпеки та забезпечує комплексне рішення для підвищення безпеки програмної системи розумної парковки. Використовуючи проміжне програмне забезпечення безпеки, набагато простіше визначити, чи справді запит був надісланий із рідної клієнтської програми розумної парковки, і перевірити, чи не є він шкідливим і чи не містить підозрілого коду. Крім того, проміжне програмне забезпечення безпеки зменшить час роботи програми, якщо запит є невідповідним або шкідливим.

Практична цінність отриманих результатів полягає у розробці підсистеми для забезпечення безпеки кіберфізичної системи для розумного паркування.

За темою дипломної роботи було взято участь у Всеукраїнській науково-практичній конференції Інформаційні технології та інженерія (IT&I-2023), м. Миколаїв та опубліковано тези у збірниках конференції.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

53. Розумне паркування в Києві URL: https://www.ukrinform.ua/rubric-kyiv/2283219-rozumne-parkuvanna-u-kievi-stali-do-roboti-persi-10-inspektoriv.html (дата звернення: 12.07.2022).

54. У Львові з'явилась перша розумна парковка URL: https://rubryka.com/2021/11/05/rozumna-parkovka-u-lvovi/ (дата звернення: 12.07.2022).

55. Інфрачервоні датчики руху: поради по використанню URL: https://res.ua/blog/infrachervoni-datchiki-ruhu-poradi-po-vikoristannju.html (дата звернення: 12.07.2022).

56. Чому датчики руху реагують на тварин та як цього уникнути URL:https://ajax.systems/ua/blog/what-is-pet-immunity-in-motion-detectors-and-how-to-use-it-correctly/ (дата звернення: 12.07.2022).

57. Відеоспостереження URL: https://uk.wikipedia.org/wiki/Відеоспостереження (дата звернення: 12.07.2022).

58. Авсієвич В.Р., Кузьмін А.А. Дослідження вразливостей системи розумної парковки та способи їх усунення. *Актуальні Проблеми Комп'ютерних Наук* (АПКН-2021), Хмельницький, Україна, 18-19 жовтня 2022. Хмельницький: ХНУ, 2022. с. 11-14.

59. Pavlova O., Kovalenko V., Hovorushchenko T., Avsiyevych V. Neural network based image recognition method for smart parking. *Comput. Syst. Inf. Technol.* 3, 2021. pp. 49–55.

60. Авсієвич В., Коваленко В. Аналіз інформаційних технологій для розумної парковки на основі штучних нейронних мереж. *Актуальні Проблеми Комп'ютерних Наук* (АПКН-2021), Хмельницький, Україна, 15-16 жовтня 2021. Хмельницький: ХНУ, 2021. С. 12-14

61. Radiuk P., Pavlova O., El Bouhissi H., Avsiyevych V., KovalenkoV. Convolutional Neural Network for Parking Slots Detection. *CEUR Workshop Proceedings*, 2022. 3156, pp. 284–293.

62. Avsiyevych V., Kovalenko V. Cyber-physical system for smart parking based on computer vision technology Black Sea Science 2022. *Proceedings of the International Competition of Student Scientific Works*. Odesa National University of Technology. Odessa: ONUT. 2022. pp. 335-346.

63. Львівська компанія SoftServe почала тестування паркінг-системи на базі машинного навчання. zaxid.net. URL: https://zaxid.net/lvivska_kompaniya_softserve_pochala_testuvannya_parking_sistemi_na_bazi_mashinnogo_navchannya_n1471000(дата звернення: 30.11.2022).

64. Паркінг в Буковелі. Трансфер до Карпат. URL:https://transferdokarpat.com.ua/articles/bukovel-vartist-poslugiparkingiv (дата звернення: 30.11.2022).

65. У Києві стартував пілот із впровадження «розумної» системи паркування. Офіційний портал Києва. URL:https://kyivcity.gov.ua/news/u_kiyevi_startuvav_pilot_iz_vprovadzhennya_rozumno_sistemi_parkuvannya/ (дата звернення: 30.11.2022).

66. Інтелектуальна система парковки Acer. URL: https://www.acer.com/ac/ru/RU/content/acerdesign-smart-parking (дата звернення: 30.11.2022).

67. Apple App Store URL:https://www.apple.com/ua/app-store/ (дата звернення: 30.11.2022).

68. Google Play Market URL:https://play.google.com/store (дата звернення: x

69. Вимоги до програмного забезпечення URL:https://uk.wikipedia.org/wiki/ Вимоги до програмного забезпечення (дата звернення: 03.12.2022).

70. Функціональні вимоги URL:https://uk.wikipedia.org/ Функціональні вимоги (дата звернення: 03.12.2022).

71. Нефункціональні вимоги URL:https://uk.wikipedia.org/wiki/ Нефункціональні вимоги (дата звернення: 03.12.2022).

72. Коваленко В. В. Кіберфізична система розумної парковки на основі технології комп'ютерного зору: кваліфікаційна робота магістра: 123 Комп'ютерна інженерія. ХНУ. Хмельницький, 2022. 113с.

73. Gollapudi S. Learn computer vision using opencv: with deep learning CNNs and RNNs : eBook. Berkeley, CA : Apress. 2019. 171 p.

74. Balaji G.V., Bharath K., Nithin S., Pranesh D.M., Shilpa S.B. Object detection using OpenCV and deep learning. *International Journal for Research in Applied Science and Engineering Technology*. 2021. Vol. 9, No. 1. pp. 3920-3923.

75. Parking Startups Are Cashing In on America's Traffic Surge URL:https://www.bloomberg.com/news/articles/2021-07-22/parking-startups-cash-in-on-america-s-post-pandemic-traffic-surge-with-apps (дата звернення: 07.12.2022)

76. Vision AI. Google Cloud. URL: https://cloud.google.com/vision (дата звернення: 02.12.2022)

77. Learn OpenCVC++ in 4 hours URL: https://www.youtube.com/watch?v=2FYm3GOonhk (дата звернення: 02.12.2022)

78. Amato G., Carrara F., Falchi F., Gennaro C., Vairo C. CNRPark+EXT. A Dataset for Visual Occupancy Detection of Parking Lots. URL:http://cnrpark.it/ (дата звернення: 07.12.2022)

79. Parking Lot Database URL: https://web.inf.ufpr.br/vri/databases/parking-lot-database/ (дата звернення: 07.12.2022)

80. Melbourne University Dataset URL: https://melbourne.figshare.com/articles/dataset/MATLABCodeCNNSVM_zip/12978932/1?file=24726374 (дата звернення: 07.12.2022)

81. Marek M. Official repository for the "Image-Based Parking Space Occupancy Classification: Dataset and Baseline" paper. URL:

https://github.com/martin-marek/parking-space-occupancy (дата звернення: 07.12.2022)

82. Detectron2 - a platform for object detection, segmentation and other visual recognition tasks. URL:https://github.com/facebookresearch/detectron2 (дата звернення: 07.12.2022)

83. Репозитарій із датасетом зображень, зібраних з камер зовнішнього спостереження ХНУ URL: https://github.com/soolstafir/khnu-parking-lot (дата звернення: 02.12.2022)

84. IP-відеокамера та комплектуючі URL: https://rozetka.com.ua/hikvision_ds_2cd2121g0_is_c_2_8_mm/p322796167/characteristics/ (дата звернення: 07.12.2022)

85. IP-відеокамера Dahua DH IPC HDW2431 URL: https://rozetka.com.ua/dahua_dh_ipc_hdw2431tp_as_s2_3_6_mm/p175174490/characteristics/ (дата звернення: 07.12.2022)

86. Келер А., Брадскі Г. Вивчення OpenCV 3: комп'ютерний зір у C++ з бібліотекою OpenCV 1st Edition. O'Reilly Media .2017 . 1024 с.

87. Кузьмін А. А., Павлова О. О. Застосування комп'ютерного зору для кіберфізичної системи розумної парковки. *"Інформаційні технології та інженерія - 2023"*, 7-10 лютого 2023, Миколаїв, Україна, с. 45-47

88. Introduction to OpenCV-Python Tutorials URL: https://docs.opencv.org/3.4/d0/de3/tutorial_py_intro.html (дата звернення: 19.02.2023)

89. How to Build an Effective API Security Strategy URL:https://www.gartner.com/en/documents/3834704 (дата звернення: 19.02.2023)

90. The PHP Framework for Web Artisans URL:https://laravel.com/ (дата звернення: 19.02.2023)

91. Laravel Framework GitHub URL: https://github.com/laravel/framework (дата звернення: 19.02.2023)

92. Laravel - Overview URL: https://www.tutorialspoint.com/laravel/laravel_overview.htm (дата звернення: 19.02.2023)

93. Hovorushchenko T., Pavlova O., Kostiuk M. Method of Increasing the Security of Smart Parking System. *JCSANDM*. vol. 12, no. 03, 2023. pp. 297–314.

94. Hovorushchenko T., Boyarchuk A., Pavlova O., Bobrovnikova K. Agent-Oriented Information Technology for Assessing the Initial Stages of the Software Life Cycle. ICTERI Workshops. 2019. pp. 617-632.

95. Understanding OWASP Mobile Top 10 Risks with Real-world Cases. URL: https://appinventiv.com/blog/owasp-mobile-top-10-real-world-cases/amp (дата звернення: 19.02.2023)

96. The top API security risks and how to mitigate them. URL: https://appinventiv.com/blog/how-to-mitigate-api-security-risks/ (дата звернення: 19.02.2023)

97. Lopatto I., Hovorushchenko T., Popov P., Pavlova O. Intelligent Multi-Agent System for Improving the Quality of Software by Taking into Account the Information of the Subject Area at All Stages of its Development. *Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. IDAACS 2021. 2021. 1. pp. 548–551.

98. Amiri W. A., Baza M., Banawan K., Mahmoud M., Alasmary W., Akkaya K. Towards Secure Smart Parking System Using Blockchain Technology. 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC). 2020. pp. 1-2. doi: 10.1109/CCNC46108.2020.9045674.

99. Waheed P., Krishna P.V. Comparing Biometric and Blockchain Security Mechanisms in Smart Parking System. *2020 International Conference on Inventive Computation Technologies* (ICICT). 2020. pp. 634-638. doi: 10.1109/ICICT48043.2020.9112483.

100. Kumar M., Khan M. H., Umar M. S. Smart parking system using RFID and GSM technology. *2017 International Conference on Multimedia, Signal Processing and Communication Technologies* (IMPACT), 2017. pp. 180-184. doi: 10.1109/MSPCT.2017.8364000.

101. Lee C., Leng FTJ., Habeeb RAA., Amanullah MAA, Rehman M., Edge computing-enabled secure and energy-efficient smart parking. *A review, Microprocessors and Microsystems*. Volume 93, 2022

102. Ahmed S., Soaibuzzaman M., Rahman S., Rahaman S. A Blockchain-Based Architecture for Integrated Smart Parking Systems. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops* (PerCom Workshops). 2019, pp. 177-182, doi: 10.1109/PERCOMW.2019.8730772.

103. Biyik M., Allam M., Pieri G., Moroni G., O'Fraifer M., O'Connell M., Olariu M., Khalid M. Smart Parking Systems: Reviewing the Literature, Architecture and *Ways Forward. Smart Cities*. 2021. 4(2). pp. 623-642.

104. Imbugwa G.B., Mazzara M. Towards a Secure Smart Parking Solution for Business Entities. *Advanced Information Networking and Applications*. AINA 2021. Lecture Notes in Networks and Systems. vol 227. Springer. pp. 469-478.

105. Abdulkader O., Bamhdi A. M., Thayananthan A., Jambi K., Alrasheedi A. A novel and secure smart parking management system (SPMS) based on integration of WSN, RFID, and IoT. *2018 15th Learning and Technology Conference (L&T)*. 2018. pp. 102-106, doi: 10.1109/LT.2018.8368492.

106. Singh SK, Pan Y, Hyuk J. Blockchain-enabled Secure Framework for Energy-Efficient Smart Parking in Sustainable City Environment. *Sustainable Cities and Society*. vol. 76. 2022.

107. Garcia M., Rose P., Sung P., El-Tawab E., Secure Smart Parking at James Madison University via the Cloud Environment (SPACE). *2016 IEEE Systems and Information Engineering Design Symposium (SIEDS)*, 2016. pp. 271-276. doi: 10.1109/SIEDS.2016.7489313.

108. Anwar A., Ijaz-ul-Haq N., Saadati P. Smart Parking: Novel Framework of Secure Smart Parking Solution using 5G Technology. *2021 IEEE International Smart Cities Conference* (ISC2). 2021, pp. 1-4. doi: 10.1109/ISC253183.2021.9562776.

109. Hovorushchenko T., Pavlova O., Avsiyevych V. Method of Assessing the Impact of External Factors on Geopositioning System Operation Using Android GPS API. *2021 International Scientific and Technical Conference on Computer Sciences and Information Technologies* (CSIT), 2021, 1. pp. 295–298

110. Waheed A., Krishna P. V., Gitanjali J., Sadoun B., Obaidat M. Learning automata and reservation based secure smart parking system. *Methodology and simulation analysis, Simulation Modelling Practice and Theory*. vol. 106. 2021.

111. Atif Y., Ding J., Jeusfeld MA., Internet of Things Approach to Cloud-based Smart Car Parking. *Procedia Computer Science*. vol. 98. 2016

112. Hakim I.M., Christover M., Jaya Marindra A.M. Implementation of an image processing based smart parking system using Haar-Cascade method. *2019 IEEE 9th Symposium on Computer Applications Industrial Electronics* (ISCAIE-2019). pp. 222–227. IEEE Inc., Penang, Malaysia, 27-28 April 2019. DOI:10.1109/ISCAIE.2019.8743906

113. Manjula G., Rajulu G.G., Anand, J.T. Thirukrishna. Implementation of smart parking application using IoT and machine learning algorithms. *Computer Networks and Inventive Communication Technologies*. Springer Singapore, Singapore. 2022. pp. 247–257. doi:10.1007/978-981-16-3728-5_18.

114. Vakula D., Kolli Y.K.. Low cost smart parking system for smart cities. *International Conference on Intelligent Sustainable Systems*, 2017. DOI: 10.1109/ISS1.2017.8389415.

115. Tkachenko R., Izonin I., Dronyuk I., Logoyda M., Tkachenko P. Recovery of missing sensor data with grnn-based cascade scheme. *International Journal of Sensors, Wireless Communications and Control*. 2021, vol. 11, no.5. pp. 531–541

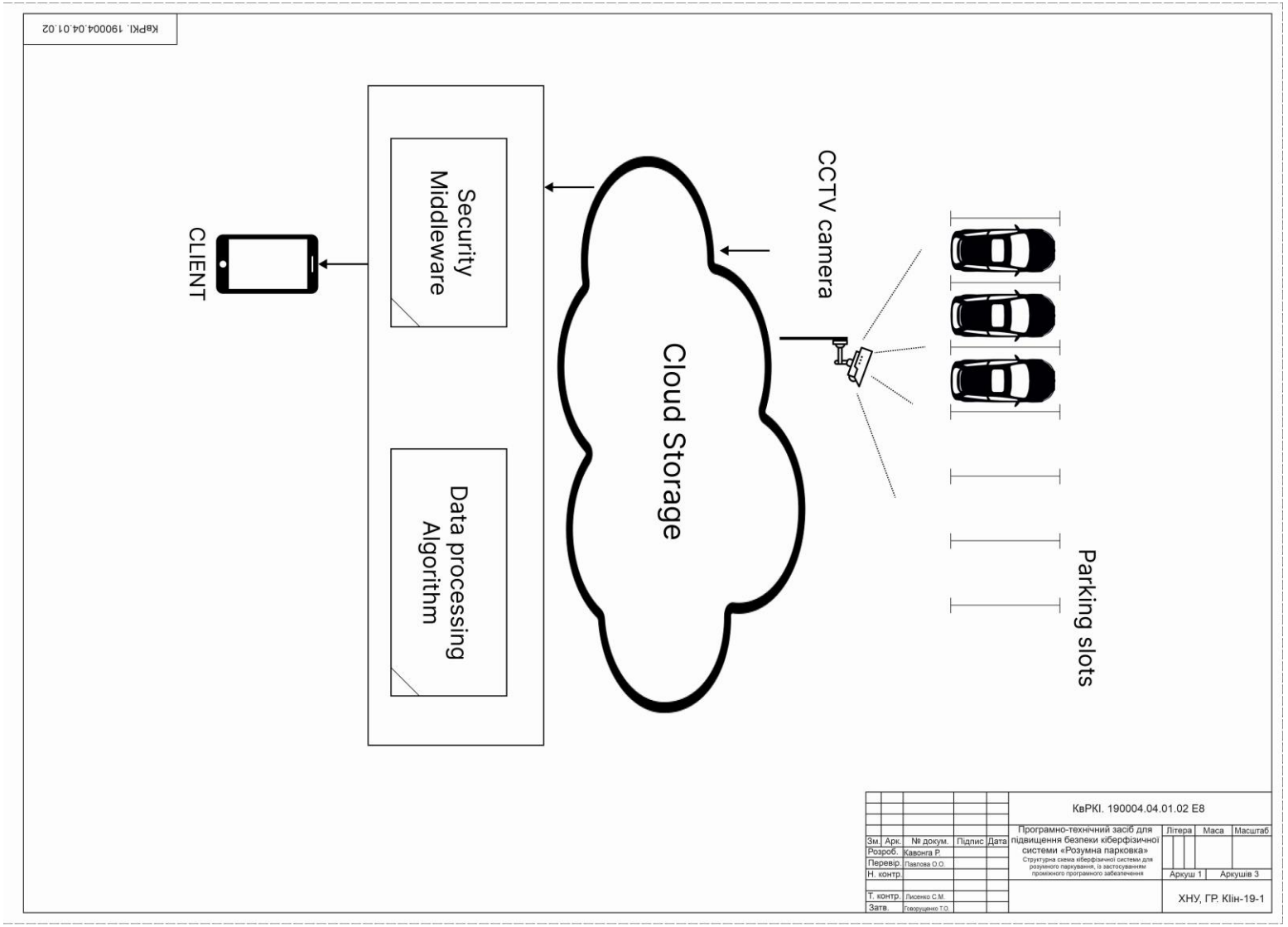
116. Zaitseva E., Levashenko V. Construction of a reliability structure function based onuncertain data. IEEE *Transactions on Reliability*. vol. 65, no. 4. 2016. pp. 1710 – 1723.

117. Zaitseva E., Levashenko V. Reliability analysis of multi-state system with application of multiple-valued logic. *International Journal of Quality and Reliability Management*. 2017. vol. 34, no. 6. P.862 – 878.

Додаток А

(обов’язковий)

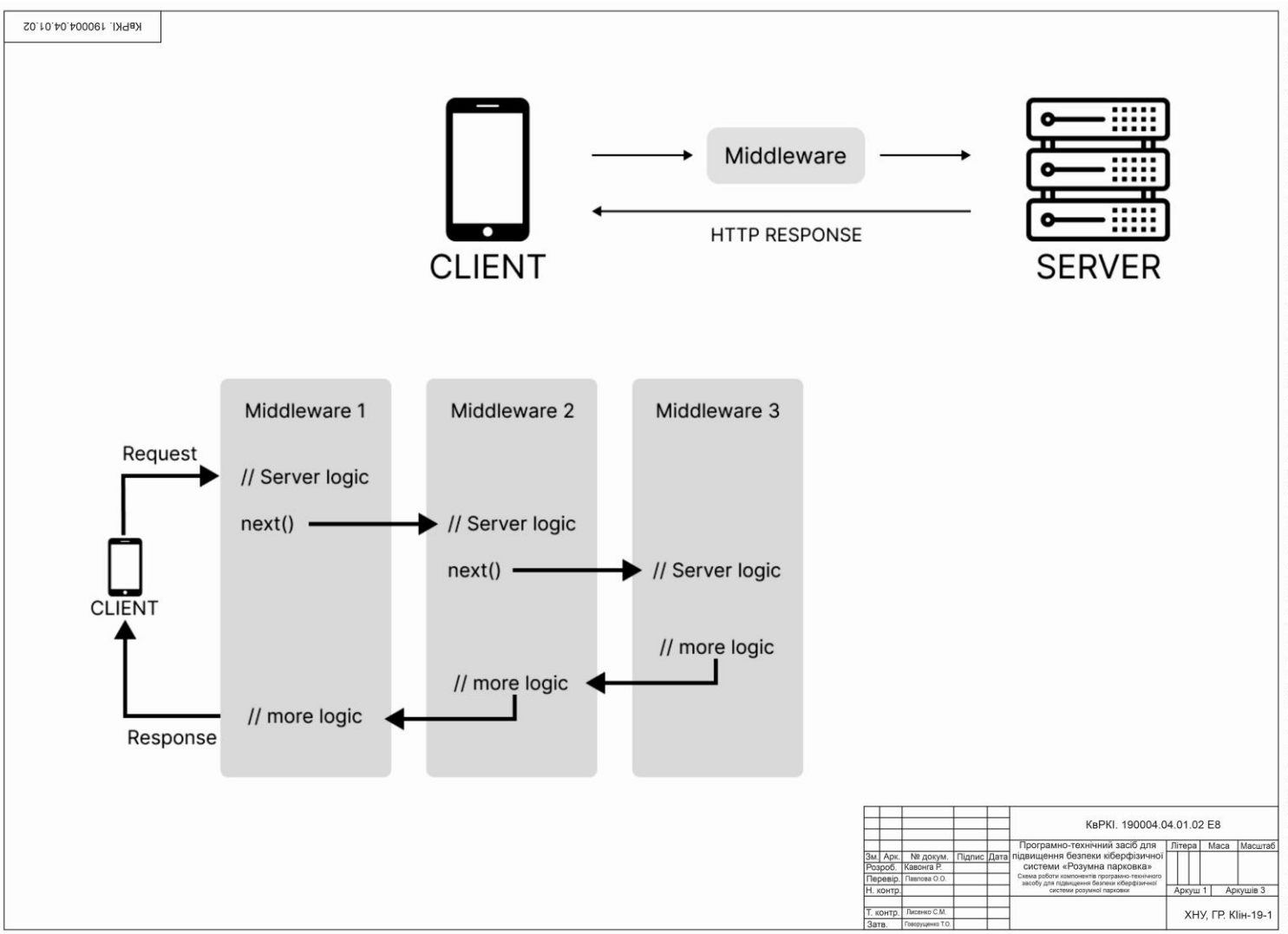
Копія креслення «Структурна схема кіберфізичної системи для розумного паркування, із застосуванням проміжного програмного забезпечення»



Додаток Б

(обов’язковий)

Копія креслення «Схема роботи компонентів програмно-технічного засобу для підвищення безпеки кіберфізичної системи розумної парковки»



Додаток В

(обов’язковий)

Копія креслення «Архітектура кіберфізичної системи для розумної парковки з використанням проміжного програмного забезпечення»

КвРКІ. 190004.04.01.02
External level
parking lane
cctv cameras
Transitional level
server
data storage
internet connection
Internal level
Security Middleware Module
Data processing Algorithm
КвРКІ. 190004.04.01.02 Е8
Програмно-технічний засіб для підвищення безпеки кіберфізичної системи «Розумна парковка»
Архітектура кіберфізичної системи для р озумної парковки з використанням промислового програмного забезпечення
Літера
Маса
Масштаб
Зм. Арк. № докум. Підпис Дата
Розроб. Кавонга Р.
Перевір. Павлова О.О.
Н. контр.
Т. контр. Лисенко С.М.
Затв. Говорущенко Т.О.
Аркуш 1
Аркушів 3
ХНУ, ГР. Кіін-19-1

Завідувачу кафедри КІІС
д-р.техн.наук, проф. Говорущенко Т. О.

Кавонга Роуз

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2ін-19-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2023 року

Підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічний засіб для підвищення безпеки кіберфізичної системи "Розумна парковка"

Автор: Кавонга Роуз

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Павлова Ольга Олександрівна, д.ф, ст.викладач

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 2,38% і адресується до 918 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи О. О. Павлова

Гарант ОП Є. Г. Лисенко

Завідувач кафедри КПС Т. О. Говорущенко

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Кавонга Роуз

Тема: Програмно-технічний засіб для підвищення безпеки кіберфізичної системи «Розумна парковка»

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень ___3____ Кількість сторінок записки ___64___

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є підвищення доступності громадських місць для людей з обмеженими можливостями.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області (проведено аналіз існуючих рішень, методів та підходів до реалізації програмно-технічних засобів для підвищення безпеки кіберфізичних систем для розумної парковки. В другому розділі кваліфікаційної роботи виконано обґрунтування вибору компонентів та середовища реалізації, а саме: апаратне середовище, функційні та нефункційні вимоги до розроблюваного програмно-технічного засобу та програмне середовище для реалізації. В третьому розділі кваліфікаційної роботи розроблено структурну схему та алгоритм роботи програмно-технічного засобу для підвищення безпеки кіберфізичної системи для розумної парковки.

4. Позитивні сторони роботи: висока актуальність та практична цінність роботи.

5. Негативні сторони роботи: недостатня увага приділена реалізації розроблюваної системи.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному інженерно-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Мартинюк Валерій Володимирович, д.т.н., професор, завідувач кафедри автоматизації та комп'ютерно-інтегрованих технологій Хмельницького національного університету

"5" 06 2023 р. _____(підпис)

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 15.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 12%**

	Документ		Сумарний збіг по Базі Даних	
ID: 114199 Назва: Програмно-технічний засіб для підвищення безпеки кіберфізичної системи «Розумна парковка» Додано в БД: 2023-05-29 Автора: Р.Кавонга Керівники: О.О. Павлова Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	77487	615	11837 (15%)	106 (17%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
114198	Назва: Software and Technical Tool for increasing the Security of Smart Parking Cyber-Physical System Додано в БД: 2023-05-29 Автора: R. Kawonga Керівники: O.O. Pavlova Консультанти: Опоненти:	11349 (15.0%)	102 (17.0%)

UNICHECK

Ім'я користувача:
Кафедра КІ

ID перевірки:
1015295495

Дата перевірки:
29.05.2023 07:45:32 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
29.05.2023 07:46:46 EEST

ID користувача:
100005591

Назва документа: **Кавонга_Програмно-технічний засіб для підвищення безпеки кіберфізичної системи «Розу...**

Кількість сторінок: **60** Кількість слів: **11228** Кількість символів: **89985** Розмір файлу: **1.81 MB** ID файлу: **1014967468**

2.81% Схожість

Найбільша схожість: **1.77%** з джерелом з Бібліотеки (ID файлу: **1014562924**)

1.44% Джерела з Інтернету	52	Сторінка 62
2.3% Джерела з Бібліотеки	49	Сторінка 62

0% Цитат

Цитати	1	Сторінка 63
Посилання	1	Сторінка 63

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	3