MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE NATIONAL AVIATION UNIVERSITY
FACULTY OF AERONAVIGATIONS, ELECTRONICS AND TELECOMMUNICATIONS
DEPARTMENT OF TELECOMMUNICATION AND RADIO ENGINEERING SYSTEMS

ADMIT TO DEFENCE
Head of the Department

____________Victor HNATIUK
"_____"______________2023 p.

QUALIFICATION WORK

(EXPLANATORY NOTE)

MASTER'S DEGREE GRADUATE

Topic: «Smart house building technologies»

Performer:______________________________ Hlib ANISIMOV
(signature)

Supervisor:______________________________ Dmytro SOLOVIOV
(signature)

Consultants from individual sections of the explanatory:

Consultant in the «Occupational Safety» Section____________ Batyr KHALMURADOV
(signature)

Consultant in the «Enviromental Protection» Section____________ Andrian JAVNIUK
(signature)

N-controller:______________________________ Denys BAKHTIIAROV
(signature)

Kyiv 2023

NATIONAL AVIATION UNIVERSITY

Faculty aeronavigations, electronics and telecommunications
Department telecommunication and radio engineering systems
Speciality 172 «Telecommunications and radio engineering»
Educational and professional program «Telecommunication systems and networks»

ADMIT
Head of the Department

____________ Victor HNATIUK
“_____” _______________2023 p.

TASK
to perform qualification work

Anisimov Hlib Evhenovich
(surname, first name, patronymic of the graduate)

1. The topic of qualification work: «Smart house building technologies»
approved by the rector’s on september «28» 2023. №1965/ст
2. The term of work: from 02.10.2023 to 31.12.2023.
3. Output to work: smart house software system
4. The content of the explanatory note: analysis of the subject field, formation of requirements for the software system, architecture and design of the software system, description of adopted software desicions, testing of the developed software
5. List of mandatory graphic (illustrative) material: developed software system application and devices

6. Calendar schedule

№ cf.	Task	The term of execution	Execution mark
1	Develop a detailed content of sections of qualification work	02.10.2023-04.10.2023	Done
2	Introduction	05.10.2023-08.10.2023	Done
3	Analysis of the subject field	09.10.2023-22.10.2023	Done
4	Formation of requirements for the software system	23.10.2023-05.11.2023	Done
5	Architecture and design of the software system	06.11.2023-15.11.2023	Done
6	Description of adopted software desicions	16.11.2023-24.11.2023	Done
7	Testing of the developed software	25.11.2023-30.11.2023	Done
6	Labor protection	01.12.2023-06.12.2023	Done
7	Enviromental protection	07.12.2023-17.12.2023	Done
8	Eliminatiion of deficiencies and defence of qualification work	18.12.2023-31.12.2023	

7. Consultants from separate sections

Chapter	Consultant (position, Full Name)	Date, signature	
		Issued the task	Task accepted
Occupational Safety	Ph.D. in Med., Professor Batyr KHALMURADOV		
Enviromental Protection	Ph.D. in Biol., Associate Professor Andrian JAVNIUK		

8. Release date of the task: September “29", 2023

Supervisor of Qualification Work __________________ Dmytro SOLOVIOV
(signature of the supervisor) (full name)

The task has been taken on for execution __________________ Hlib ANISIMOV
(graduate signature) (full name)

ABSTRACT

Qualification work «Smart house building technologies» contans _93_ pages, _33_ figures, _2_ tables, _21_ used sources.

IOT, NFC, BLUETOOTH, WIFI, SERVERLESS, SECURITY, SWIFT, IOS, IPADOS, CORE NFC, CORE BLUETOOTH, SWIFTUI, SWIFT DATA RASPBERRY PI, PYTHON, BLUEPY, NFCPY, SMART HOME.

Object of study – smart house building technologies.

Subject of study – the architecture of building a smart house in the absence of server dependency.

The purpose of qualifcation work – the purpose of the qualification work is to study the technologies of building smart houses and develop the architecture of a smart house in the absence of server dependence.

Research method – the following methods were used to solve the set of tasks: data aggregation with confidentiality, stability of quantum computing, coding and information theory, secure NFC communication, machine learning to detect anomalies, energy-efficient communication protocols, number theory approaches, human-computer interaction research computer, development of distributed systems, dynamic key management systems.

It is recommended to use qualification work materials when creating an autonomous smart home, or improving the protection of the already existing.

CONTENT

INTRODUCTION

Topic relevance. IoT, or the Internet of Things, refers to a network of interconnected physical devices, vehicles, appliances, and other objects with built-in sensors, software, and connectivity that enable them to collect and share data over the Internet. Often referred to as "smart" devices, these devices can share data with each other and with cloud applications to provide various functions, automation and data-driven insights.

IoT devices are equipped with different communication technologies such as Wi-Fi, Bluetooth, Zigbee, cellular networks or even low-power protocols such as LoRaWAN or NB-IoT. These technologies allow devices to connect to each other, the Internet, or a local area network, forming a network of interconnected devices.

Overall, IoT has the potential to revolutionize various industries by bridging the physical and digital worlds, bringing automation, efficiency and new opportunities to businesses and individuals. However, it also presents scalability, security, interoperability, and data management challenges that must be addressed to fully realize its benefits.

The aim and objectives of the research. The purpose of this study is to review smart house building technologies and identify problems and actualize solutions in this area.

To achieve the set goal, the following scientific tasks are solved:

1. A review of the literature on technologies for building smart houses.
2. Analysis of existing architectures of smart home systems.
3. Development of smart home architecture and improvements to existing ones to increase autonomy and security.

The object of research. smart house building technologies.

The subject of research. the architecture of building a smart house in the absence of dependence on the server.

Research methods. Privacy-preserving data aggregation, resilience of quantum computing, coding and information theory, secure NFC communication, machine learning for anomaly detection, energy-efficient communication protocols, number theory approaches, human-computer interaction research, distributed systems design, dynamic key management systems.

Practical significance of the obtained results.

The obtained results of building a smart house in the absence of dependence on the server can be of practical importance for the development and implementation of similar networks in various fields, such as industry, social infrastructure, home automation, military, and others.

In particular, the research results can help in the design of an autonomous smart house, as well as in the creation of an autonomous surveillance system.

In addition, the research results can be useful for improving the security of existing smart homes.

Thus, the obtained results of building a smart house in the absence of dependence on the server can be important for the development and implementation of similar networks in various industries.

CHAPTER 1
ANALYSIS OF THE SUBJECT FIELD

1.1. Analysis of the subject field

Smart homes are a revolutionary integration of technology into people's everyday lives, transforming traditional living spaces into connected and intelligent environments. At the heart of the smart home is the implementation of the Internet of Things (IoT), where devices and appliances are interconnected through a network, allowing for seamless communication and automation. Homeowners can remotely monitor and control various aspects of their homes using smartphones or voice-activated assistants, increasing convenience, energy efficiency and security.

One of the key aspects of smart homes is home automation, which allows for centralized control of lighting, heating, ventilation, air conditioning and other appliances. This not only streamlines daily tasks, but also helps conserve energy by optimizing resource usage based on real-time conditions. For example, smart thermostats can learn user preferences and adjust the temperature accordingly, resulting in energy savings and increased comfort. In addition, smart lighting systems can be programmed to adapt to different scenarios, promoting energy-efficient practices.

Security is another primary concern that smart home technologies address. Integrated surveillance cameras, door cameras and smart locks provide homeowners with monitoring and control over their property. In addition, smart security systems can send instant alerts in the event of suspicious activity or a breach, offering an additional layer of protection. As smart home technology continues to advance, the potential for customization and integration of new devices is expanding, promising an even more connected and intelligent living experience in the future.

NFC (Near Field Communication) and Bluetooth are wireless communication technologies that can be used in the context of the Internet of Things (IoT) to connect and exchange data between devices. When it comes to a serverless solution in IoT, both NFC and Bluetooth have their respective applications and advantages [1].

NFC in IoT: NFC is a short-range wireless communication technology that works over a distance of several centimeters. This allows two devices to establish a connection by bringing them closer to each other. NFC is commonly used for contactless payments, access control systems and data exchange between devices.

In the IoT context, NFC can be used for a variety of serverless applications such as:

• Connecting devices: NFC simplifies the process of connecting devices in the IoT ecosystem. By touching an NFC-enabled device to an IoT device, they can establish a secure connection without the need for complex configuration.

• Data transfer: NFC can facilitate the transfer of small amounts of data between devices. For example, an NFC tag embedded in a product can provide product details, inventory information, or trigger an action when scanned by an NFC-enabled device.

• Authentication and security: NFC can improve security in IoT systems. For example, NFC can be used for authentication, allowing only authorized devices to interact with each other.

Bluetooth in IoT: Bluetooth is a widespread wireless communication standard that works over short distances, typically up to 100 meters. It provides a reliable and energy-efficient connection between devices. Bluetooth is widely used in audio devices, portable devices and home automation systems [2].

In the serverless realm of IoT, Bluetooth offers several advantages:

• Device connectivity: Bluetooth enables seamless connectivity of devices in the IoT ecosystem without the need for a central server. Devices can communicate with each other directly or form a mesh network that allows data sharing and control without relying on a central server.

• Energy efficiency: Bluetooth Low Energy (BLE) is an energy efficient version of Bluetooth designed for IoT applications. BLE devices consume minimal power, making them suitable for battery-powered IoT devices and sensors [3].

• Proximity-based interactions: Bluetooth's range and ability to detect nearby devices make it suitable for proximity-based interactions. For example, a Bluetooth-enabled beacon can trigger certain actions or provide location information when a user's device comes into range.

Solutions with servers in the architecture are not always reliable - because the server may fail, or it may be exposed to an attack, as a result of which user data will be stolen, or a third party will gain access to the IoT device [4]. An example of the interaction between the components of such an architecture is shown in Fig. 1.1.

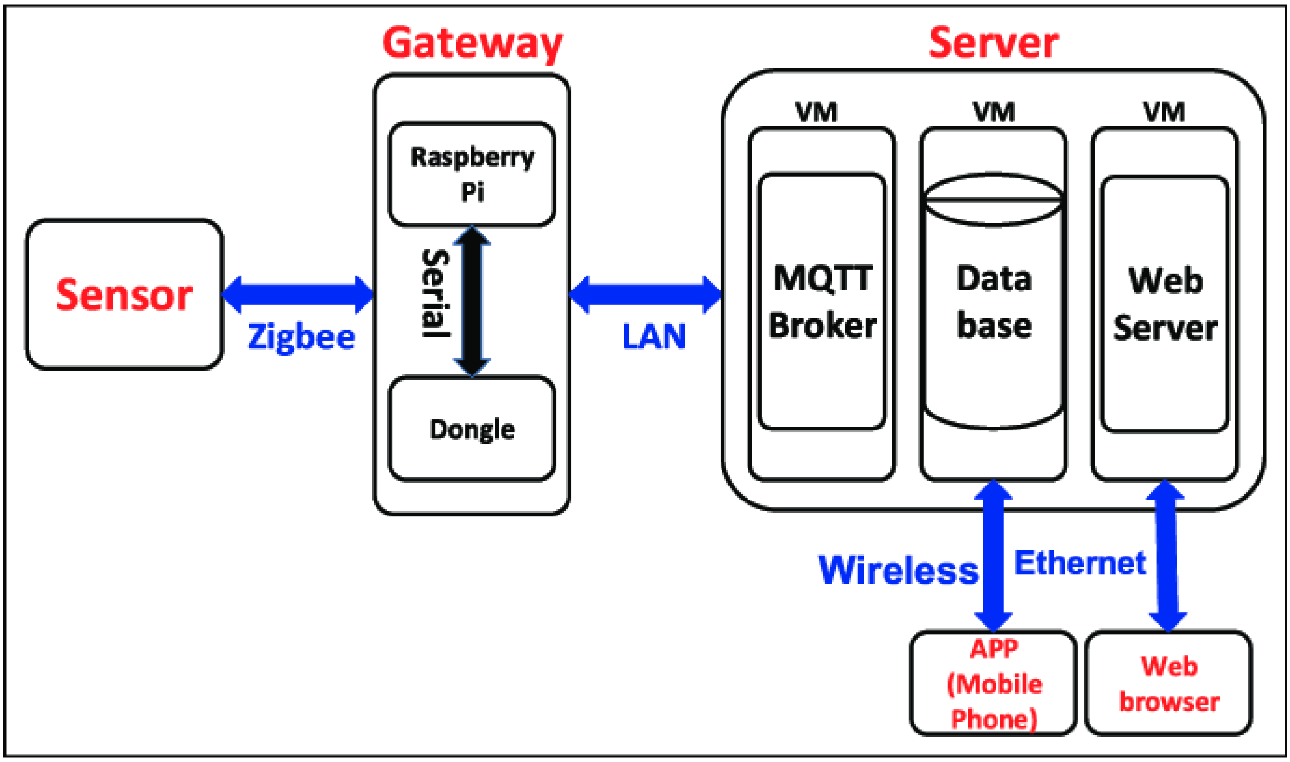

Fig. 1.1 IoT based smart home system with sensor, gateway, server, web browser and mobile app

Serverless solution in NFC and Bluetooth IoT: In a serverless IoT architecture, NFC and Bluetooth technologies can be used to establish direct peer-to-peer communication between devices without relying on a central server to process or store data. This approach offers benefits such as reduced latency, improved privacy, and increased reliability. Using NFC and Bluetooth, devices can interact, share data and perform tasks autonomously, contributing to a decentralized and efficient IoT ecosystem. An example of the main idea of serverless architecture is shown in Fig. 1.2.

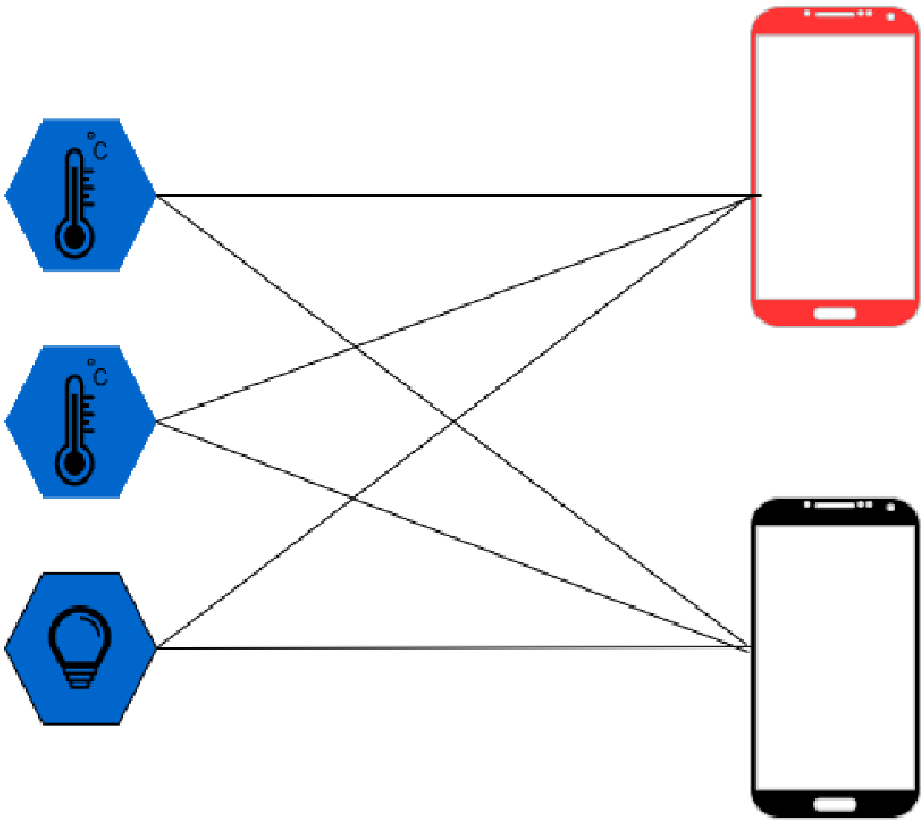

Fig. 1.2. Serverless version of smart home architecture

However, it's worth noting that while NFC and Bluetooth are suitable for certain IoT use cases, they may not be ideal for scenarios involving long-distance communication or large-scale deployment. In such cases, other wireless technologies such as Wi-Fi, cellular networks, or even hybrid approaches that combine multiple connectivity options may be more appropriate.

A serverless approach in IoT offers several advantages that contribute to the efficiency and scalability of IoT systems. Here are some key benefits of using a serverless architecture in the IoT context:

1. Reduced infrastructure complexity: In a serverless approach, the responsibility for managing and provisioning servers is transferred to the cloud service provider. This

eliminates the need for the IoT device owner to configure and maintain a complex server infrastructure. As a result, this reduces operational overhead and allows the focus to shift to the development and deployment of IoT applications.

2. Cost optimization. With a serverless approach, you only pay for the resources you consume, usually per execution. This eliminates the need to provision and pay for idle resources as the cloud provider dynamically scales the infrastructure based on incoming workload. This model can be particularly beneficial for IoT deployments that have varying levels of usage or intermittent communication requirements.

3. Scalability and Elasticity: Serverless architectures offer automatic scaling and elasticity. Cloud providers handle resource scaling in response to demand, ensuring that IoT applications can adapt to fluctuations in usage without manual intervention. This capability is critical for IoT systems that experience bursts of traffic or sudden increases in the number of connected devices.

4. Reduced development time: Serverless platforms abstract away the underlying infrastructure, forcing developers to focus solely on writing application logic. This results in faster development cycles and shorter time-to-market for IoT solutions. Developers can use pre-built features and services provided by the cloud provider, which can significantly speed up development and deployment processes.

5. Simplified maintenance and upgrades: With a serverless approach, the cloud provider takes care of infrastructure maintenance, security patches, and upgrades. This relieves the IoT device owner of the burden of managing and maintaining servers, allowing them to focus on the core functions of their IoT application. This ensures that the system stays up-to-date with the latest security patches and enhancements without requiring manual intervention.

6. Improved scalability: Serverless architectures inherently support horizontal scaling, allowing IoT applications to handle increasing numbers of devices and requests without manual configuration. This scalability is critical for IoT deployments, where the number of connected devices can grow rapidly over time.

7. Higher availability and fault tolerance: Cloud providers typically offer high availability and fault tolerance in their serverless offerings. Core infrastructure is distributed and replicated across multiple data centers, reducing the risk of single points of failure. In the event of an outage or failure, the cloud provider automatically redirects workloads to available resources, minimizing downtime and ensuring continuity of IoT applications.

While a serverless approach to the Internet of Things offers several advantages, it is important to be aware of the potential challenges and issues that may arise. Here are some common problems associated with a serverless approach:

1. Cold start latency: In a serverless architecture, functions or services are typically called on demand in response to an event. However, if the function is not yet running or "warmed up", it can be a cold start. A cold start occurs when the cloud provider needs to allocate resources and initialize a feature, resulting in increased latency for the first request. This can be problematic for real-time or time-sensitive IoT applications that require low-latency responses [5].

2. Limited execution time and resource limitations: Serverless functions often have limitations in terms of execution time and available resources. Cloud providers usually set a maximum execution time for functions, which can range from a few seconds to a few minutes. In addition, there may be limits on the amount of memory or storage that a function can use. These limitations can affect the design and execution of IoT applications, especially those with complex or time-consuming operations.

3. Vendor lock-in: Taking a serverless approach can lead to vendor lock-in, as each cloud provider has its own unique serverless platform and APIs. If an IoT solution relies heavily on the services and functions of cloud providers, it may be difficult to switch to another provider in the future. It is important to carefully consider the long-term implications and assess the tolerability of the solution to reduce the risks associated with vendor lock-in.

4. Lack of control and visibility: In a serverless approach, the underlying infrastructure and resource management is handled by the cloud provider. This can lead to

a loss of control and visibility of the underlying infrastructure, making it difficult to troubleshoot, monitor performance, or optimize resource allocation. It is critical to ensure that the cloud service provider offers sufficient monitoring and debugging tools to mitigate these issues.

5. Security and Compatibility Issues: Serverless architectures introduce additional security aspects. Because the runtime is shared by multiple functions, there is a risk of data leakage or cross-functional vulnerability. To protect sensitive IoT data, it is important to carefully design and implement security measures such as proper authentication, authorization, and data encryption. Compliance with industry regulations and data protection laws can also be challenging in a serverless environment.

6. Cost Management: While serverless architectures can provide cost optimization, they can also lead to unexpected costs if not properly managed. Features triggered by a large number of events, or those that require resources, can lead to increased costs. In addition, constant monitoring and debugging of functions can increase the total cost. It is important to carefully monitor and optimize the use of resources to avoid unexpected financial consequences.

7. Dependence on third-party services: Serverless architectures often rely on various third-party services, such as databases, message queues, or authentication services, to extend their functionality. While these services can speed up development, they create a dependency on external vendors. If a third-party service fails or changes its API, the availability and functionality of the IoT application may be affected.

It is important to note that while a serverless approach offers significant advantages, it may not be suitable for all IoT use cases. The choice of architecture can be influenced by factors such as latency requirements, data privacy and security, and regulatory compliance. Therefore, it is imperative to carefully evaluate the specific requirements and constraints of your IoT deployment before deciding on a serverless approach.

1.2. Identification of problems and actualization of solutions

As a result of the study of the subject field, its development and prospects, analysis of the serverless approach, a number of common problems that users need to solve. Most of these problems are inherent not only to conventional IoT solutions, but also to more recent solutions, such as products from Xiaomi, which use modern technologies in their quests.

1.2.1. Target audience

From the domain analysis, it can be concluded that the target audience that is interested in deploying IoT using Bluetooth and NFC technologies with a serverless approach:

- Enterprises and Business: Enterprises in various industries can benefit from using Bluetooth and NFC technologies with a serverless approach to IoT deployment. For example, retail stores can use Bluetooth beacons and NFC tags to improve customer interactions and optimize inventory management. The serverless approach allows companies to centrally manage and analyze data from these devices, enabling personalized marketing campaigns, real-time inventory updates and efficient supply chain management.

- Smart home and consumer electronics: The target audience may be consumers who are interested in building smart homes or using IoT devices in their daily lives. Bluetooth and NFC technologies can provide seamless connectivity and automation between various smart devices, such as smart thermostats, security systems, and home appliances. With a serverless approach, consumers can use cloud services to remotely manage their devices, receive alerts and notifications, and access advanced features such as power optimization and usage analytics.

- Providers of medical services and manufacturers of medical equipment. The healthcare industry can benefit from using Bluetooth and NFC technologies with a serverless approach to enable remote patient monitoring, wearable health trackers and connected medical devices. For example, healthcare providers can use Bluetooth-enabled devices to monitor vital signs, and NFC can be used to securely control access to medical records. Serverless platforms can help securely process and store patient data in compliance with privacy regulations, provide real-time alerts to healthcare providers, and enable data-driven diagnosis and treatment.

- Industrial and manufacturing sector: Industries such as manufacturing, logistics and supply chain management can use Bluetooth and NFC technologies through a serverless approach to improve operational efficiency and automation. For example, Bluetooth sensors can be used for asset tracking, equipment monitoring and predictive maintenance, while NFC tags can provide secure access control and data collection. Using a serverless architecture, industrial organizations can centrally manage and analyze sensor data, optimize processes and ensure seamless integration with back-end systems to improve productivity and save money.

- Smart cities and infrastructure: Municipalities and urban planners interested in creating smart cities and improving infrastructure can be the target audience for IoT deployments using Bluetooth and NFC technologies with a serverless approach. These technologies can be used for applications such as smart parking systems, public transport tracking and waste management solutions. A serverless approach allows for centralized management and analysis of data from various IoT devices deployed throughout the city, facilitating data-based decision-making, optimizing resources, and improving services for citizens [5].

1.2.2. General problems of the subject area

Here are some general problems of the subject area:

Data processing and storage. Bluetooth and NFC devices often have limited processing power and memory. In scenarios where significant data processing or storage is required, a serverless architecture can offload these tasks to cloud services. Using a serverless model, IoT devices can transmit raw data to the cloud for processing, analytics, and long-term storage, offloading the burden on the devices themselves.

Device management and updates. Managing a large number of IoT devices that use Bluetooth or NFC can be a challenge. The serverless approach allows for centralized device management, enabling wireless firmware updates, configuration changes, and remote monitoring. Using serverless platforms, device owners can more effectively manage and monitor their IoT deployments, ensuring that devices are updated and functioning optimally [4].

Scalability and resource management: IoT deployments often require scaling up or down based on demand. With a serverless approach, cloud providers handle dynamic scaling of resources in response to incoming workloads. This ensures that IoT applications using Bluetooth or NFC can seamlessly adapt to fluctuations in the number of connected devices or data processing requirements. Serverless architectures provide elastic scaling, improving the overall scalability and resource management of IoT deployments.

Internal integration and compatibility. Integrating Bluetooth or NFC devices with internal systems and services can be complex. Serverless architectures provide a level of abstraction that simplifies integration with various back-end services and APIs. With serverless features and services, IoT devices using Bluetooth or NFC can easily connect to other cloud systems, databases or third-party services, facilitating interoperability and improving the overall functionality of the IoT solution.

Analytics and Statistics: Bluetooth and NFC devices generate vast amounts of data that can provide valuable insights for businesses and individuals. With a serverless approach, data collected by IoT devices can be processed, analyzed and transformed into meaningful information using cloud-based analytics services. It enables real-time analytics, anomaly detection, predictive modeling, and other advanced data processing techniques that can drive powerful insights and improve decision-making.

1.3. Formulation of the problem

Smart homes are becoming more popular and widespread over time (according to Statista[6], a data collection and visualization site, by 2023 there will be more than 300 million households using smart home systems, and this number is projected to increase double by 2028).

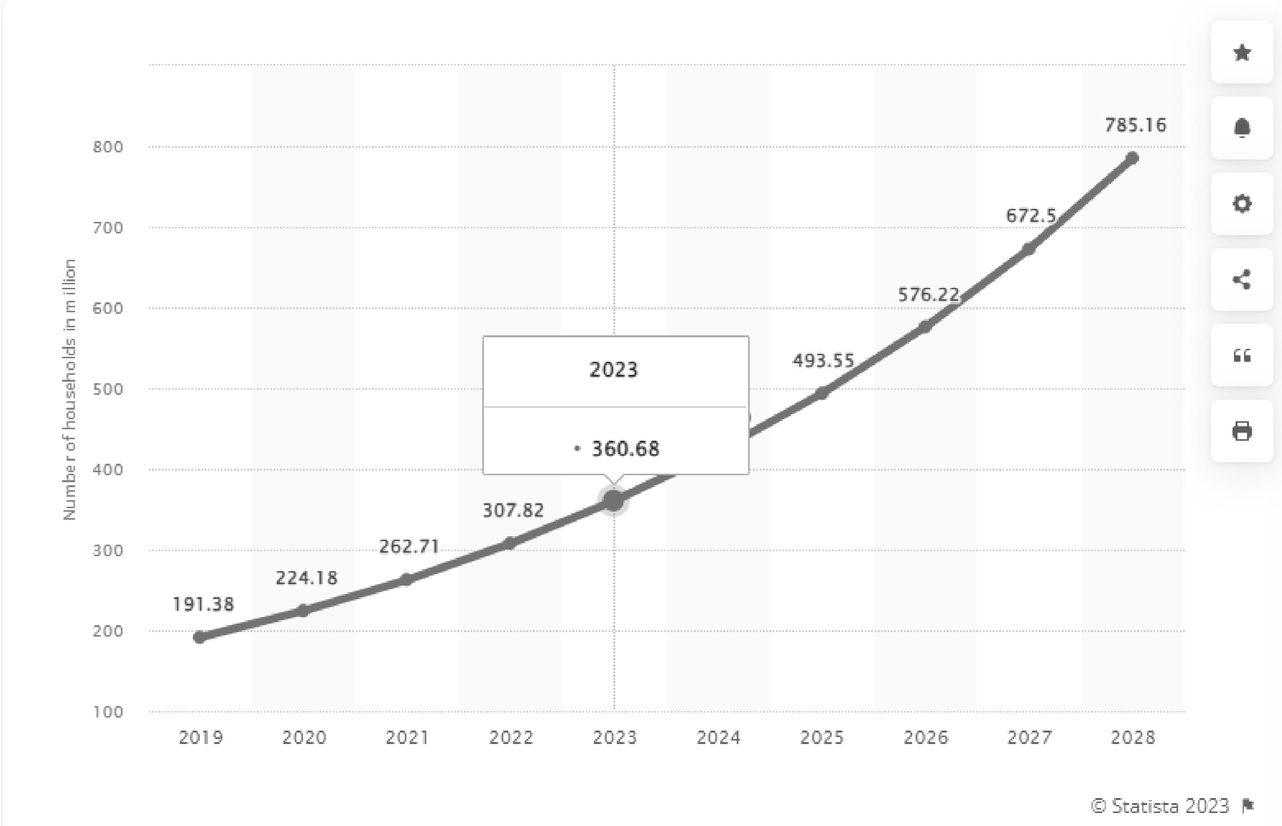

Fig. 1.3. Number of smart home users worldwide from 2019 to 2028

Considering that the use of such systems in the market is growing, there is a need to develop less costly and more efficient systems, which, at the same time, can maintain an adequate level of security and stability, the possibility of scaling and expansion, and sufficient performance.

Classic smart home management systems based on server architecture, although they have a number of advantages, such as:

- estimated price - calculated depending on the equipment and required bandwidth;

- data autonomy - in the server model, you can clearly establish where the data is and have full control over it, which allows you to better manage confidential information and comply with relevant laws;

- ease of debugging and monitoring - server models often have a pre-configured process for finding and fixing problems, collecting metrics, logging and tracking statuses, which makes it easier to track and isolate errors in the system;

- availability and unlimited runtime - a dedicated server is designed to be always online and does not require a cold start and has no time limit.

However, such systems have their drawbacks, the key of which are:

- difficulties in management and configuration - maintaining the environment can be time-consuming and expensive (routine system updates, detection and correction of technical problems in various components, including equipment, security patches, etc.);

- higher price - even when the servers are idle, you need to pay for them;

- energy consumption - physical servers consume electricity and cannot dynamically redistribute resources as needed;

- insufficient geographical coverage - if the devices included in the IoT network are geographically dispersed, or the home system has to transmit data to the mobile application when the user is not at home, remote servers may work with a delay;

- a single centralized point of vulnerability - in the absence of proper backup or clustering of the system, any error in the operation of equipment or software can lead to the failure of the entire smart home system;

- potentially complicated recovery procedure after a disaster - also in the case of physical damage (floods, fires, etc.) to a dedicated server, system recovery can take a lot of time and resources, especially in the absence of prepared backup and recovery strategies;

- limited flexibility and complexity in expansion - architectures based on the server model usually require pre-prepared decisions about the type of servers, their number and capacity, changing these settings can be quite burdensome.

The task of this work is to create a concept of a serverless architecture of a smart house based on an IoT network. To achieve the goal, it is necessary to develop a prototype of the application that will support this configuration and select the appropriate devices with which the system can be tested.

The key tasks are implementing a successful transition from a server model to a serverless one, choosing the most suitable service provider[7], implementing state management between function calls, effective cold start processing, maintaining data security and privacy, monitoring and debugging the environment.

The application developed in this work should provide real-time responses, automatic network expansion to handle different device load levels, improved cost-effectiveness by turning off unused resources, and a simplified system configuration process.

This will also include the integration and testing of several IoT devices, such as security systems, HVAC control systems, lighting systems, household appliances, sensors for explosions, radiation and air pollution, etc. It will be necessary to make sure of their compatibility and quality of response within the framework of a serverless network.

The global goal is to use a serverless architecture to build a smart home setup that is not only technically advanced, but economical, sustainable, easy to use and maintain, and flexible.

CHAPTER 2
FORMATION OF REQUIREMENTS FOR THE SOFTWARE SYSTEM

2.1. Formation of software requirements

As mentioned earlier, the goal of this work will be to build a smart home system based on the serverless architecture model. The main focus and primary task will be the creation of a mobile application that can support this concept and effectively manage a network of IoT devices. Next, the key functional and non-functional requirements for the application itself and the system as a whole will be given, on which further work will be based.

The main functionality of the application should include:

- the possibility of creating systems - by system we mean a separate configured system of a smart home, which can consist of one or more groups;

- the possibility of creating groups within systems - a group is a logical group of devices separated for convenience, for example, a separate room or a set of devices similar in functionality and use;

- the ability to connect devices and add them to groups;

- the ability to view statistics by system, group and individual device (values of data from devices by time, number of certain values);

- the possibility of personalizing the system, group and device (setting a name, choosing a separate color and photo);

- the ability to receive data from devices on demand;

- the possibility of changing the language of the application.

In addition, the devices that will be included in the system must also meet certain requirements to ensure compatibility.

The main functionality of the device, which will be used in a serverless configuration, should include:

- possibility of adjustment through the application;

- the possibility of transferring data to the application;

- the possibility of creating a pair with other devices;

- the ability to transfer data to other devices from the created pairs.

Also, if there is additional time and opportunities for development, the system may include additional functions:

- autonomy of the devices used in the system - operation without an application, the possibility of setting directly on the devices themselves within the network in full or in part (depending on the type of device);

- the possibility of remote connection to the system via Ethernet or WiFi;

- the ability to remotely control the system (for example, turn on or off the lights at home);

- automation of processes within the system - creation of routines, scripts, etc.;

- the ability to share control of the system with another person using the application.

2.2. Non-functional system requirements

The system, the prototype of which is implemented in this work, must meet not only the selected functional needs, which are directly responsible for its capabilities and configuration, but also non-functional ones, because only thanks to them can maximum efficiency, convenience and ease of use, and competitiveness be achieved.

After a detailed review and analysis of the subject area and the construction of similar settings[8], the following requirements were identified:

- scalability - ensuring automatic expansion or contraction on demand and appropriate management of available resources (for example, the system should adapt automatically, without manual intervention, when a new device is added to the IoT);

- high level of security - protection against possible unauthorized access, processing of potentially vulnerable places in the network, ensuring proper protection of private data exchanged between devices in the smart home network;

- ease of use - the architecture should be built in such a way as to simplify the management of a smart home as much as possible (have automatic software updates, be easy to configure, integration with IoT devices should be easy and not require a large number of steps);

- resistance to network errors - the ability to automatically recover after a temporary loss of communication;
- compatibility with various devices and standards;
- localization - the ability to use the application with different languages, in addition, compliance with data regulations specific to the region of use;
- maintaining data integrity and preserving their privacy;
- availability - the system must be designed taking into account the possible almost constant working time and ensure uninterrupted execution of daily tasks;
- high performance - even under high load, the system must respond with minimal delay, especially considering that a large number of smart home functions are critical and urgent (for example, security systems);
- stable connection between the application and recipients;
- cost-effectiveness - estimated cost and preservation of proportionality between needs and real use.

CHAPTER 3
ARCHITECTURE AND DESIGN OF THE SOFTWARE SYSTEM

3.1. UML designing

We will design the architecture of the application and the functions it will perform in more detail by modeling the developed application using UML diagrams.

For qualitative modeling of the system, it is necessary to choose the boundaries of the projected system, the main actors and the functionality that can be available to these actors.

The user who will use the system has a need to manage smart systems, groups, devices, receive data updates in real time.

In the event of a device change, for example, failure, it is necessary to be able to transfer data.

Due to the possible amount of data in the system, the user should see statistics by system, group, device for convenient visualization and perception of data.

For the convenience of the user, there should be a quick setting of the device, and the ability to receive the data of the entire system from one point.

To achieve the goal of the system, it will be necessary to design the UML diagram of the application and the device in the system.

A use case diagram shows all possible ways of using an application at the highest level of abstraction.

Thus, the main functionality of the user is to work with the application. The main actor is the application user. The designed diagram for the smart serverless home usage and management application is shown below in Figure 3.1.

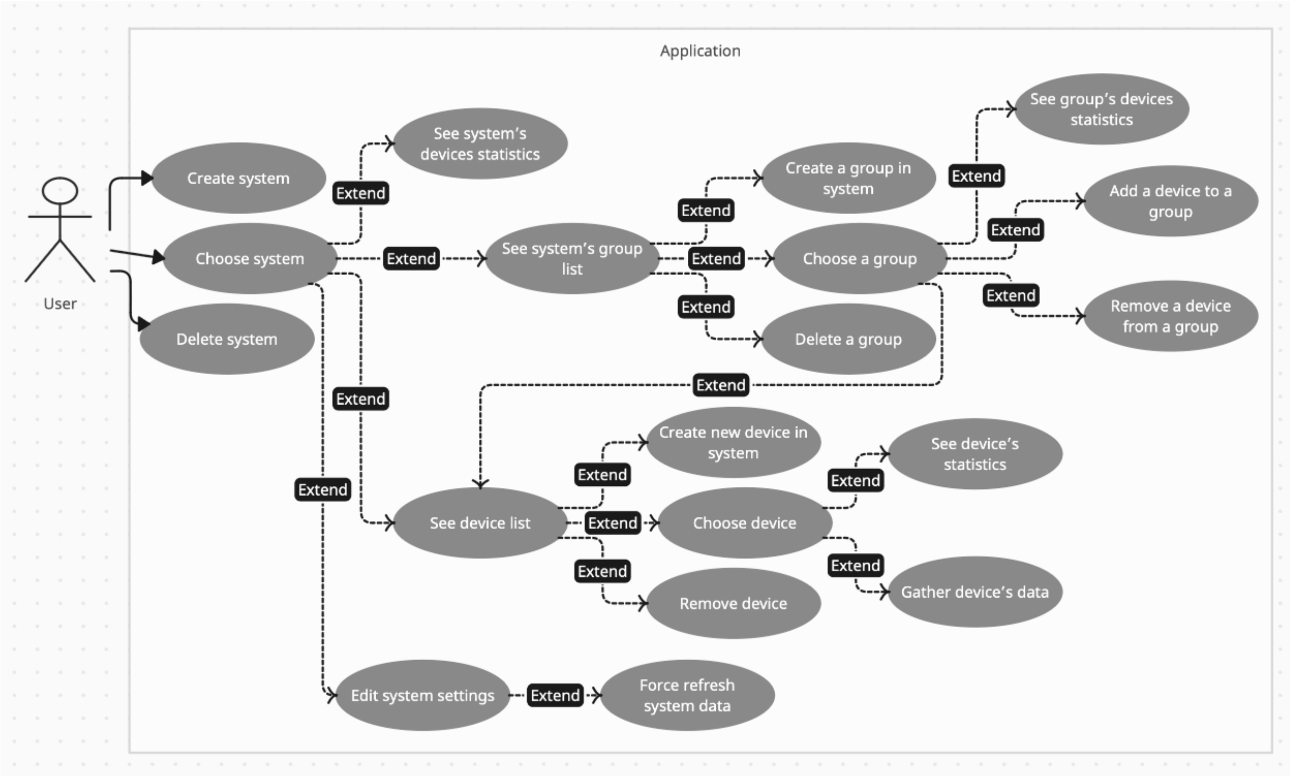

Fig. 3.1. Use Case diagram of the application

The main functionality of the device is data transfer with the application and other devices. The main actors are an app and another device. The designed diagram for the device is shown below in Figure 3.2.

For the convenience of the user, the device has been designed with a view to the fact that the user can get the data of all the devices within reach through one, by creating pairs.

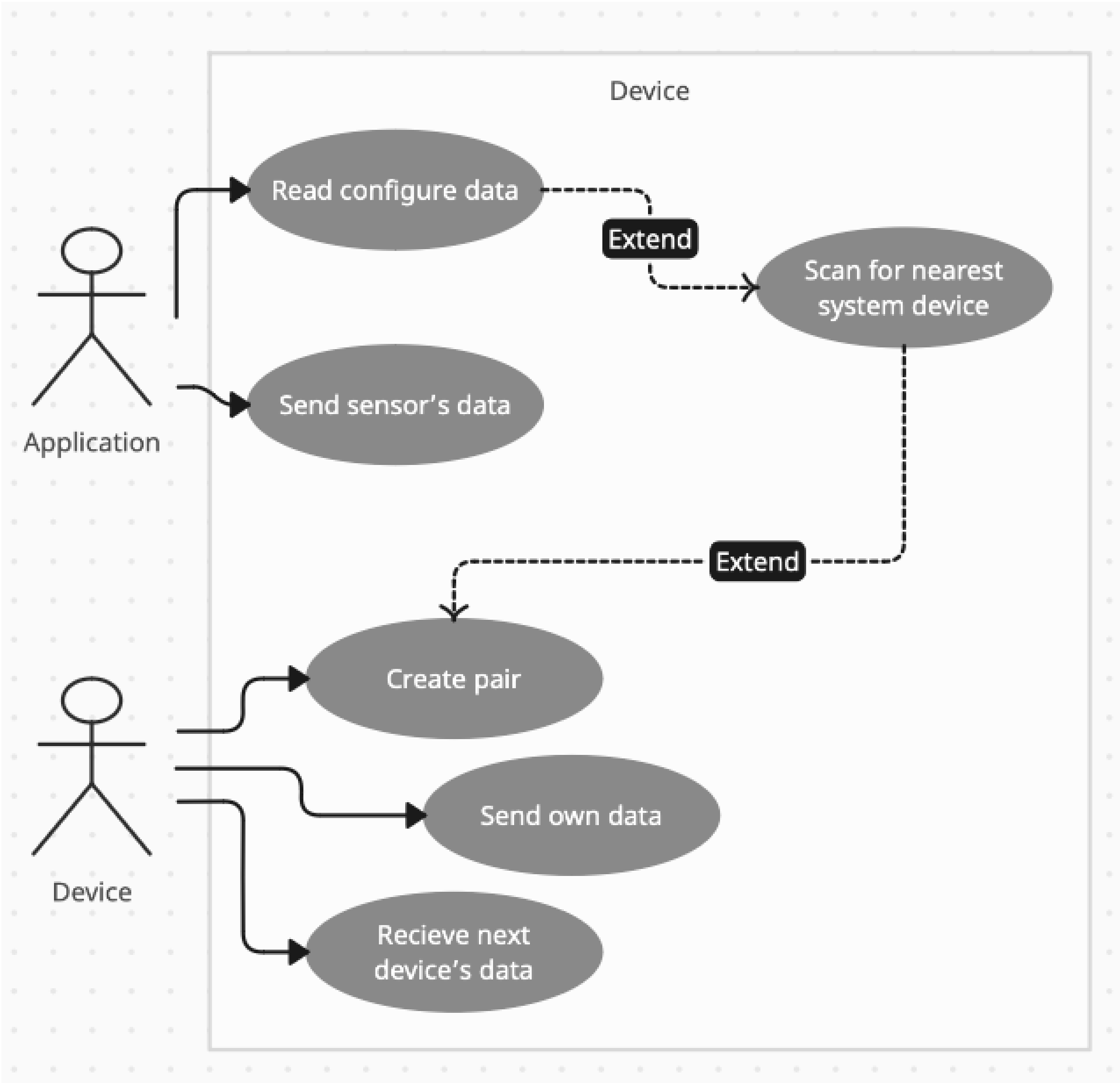

Fig. 3.2. Use Case diagram of the device

For added safety and ease of use, the device has been designed with ease of use, setup and operation in mind.

3.2. Software architecture design

The application's architecture follows the MVVM pattern, facilitated by the carefully chosen frameworks for the user interface. In opting for SwiftUI over UIKit for the iOS app, the decision was motivated by UIKit's imperative, event-driven nature. In UIKit, each view in the hierarchy could be referenced, and updates were made when the view loaded or in response to events like button clicks or data updates for display in UITableView. Event handling involved callbacks, delegates, and target actions, which were managed with the Combine framework. However, in SwiftUI, the approach shifts to a declarative and managed framework.

SwiftUI operates in a declarative manner, eliminating the ability to reference views directly or alter them in response to events. Instead, changes are made to the state bound to the view. Traditional event-handling mechanisms such as delegates, target actions, and the chain of responders are replaced with closures and bindings.

In SwiftUI, a View is viewed as a fundamental programming feature. It takes an input (state) and produces an output, with modifications to the output achieved by changing the input. Unlike UIKit, where views could be dynamically manipulated at runtime, SwiftUI enforces a more static structure. Any adjustments to the displayed UI must be explicitly declared within the body function and cannot be altered dynamically.

The decision was consciously made to leverage the native Core Bluetooth framework for establishing robust Bluetooth communication. This strategic choice provides the advantage, in stark contrast to third-party frameworks, of seamlessly enabling the implementation of essential tasks using cutting-edge technologies that seamlessly integrate into the operating system. Bluetooth, a widely adopted short-range wireless technology standard, facilitates efficient data exchange between fixed and mobile devices over limited distances, fostering the creation of reliable personal networks. Core Bluetooth, specifically designed for this purpose, furnishes the essential classes and tools for applications to effortlessly communicate with Bluetooth-enabled low-energy, base-rate, and high-data-rate wireless technologies.

NFC technology was strategically chosen for device setup due to its limited operational range, enhancing security by necessitating close proximity for access. Additionally, NFC streamlines the development of both applications and devices by eliminating the need for a separate network configuration, thereby bolstering energy and cost efficiency.

NFC, comprising communication protocols facilitating interaction within a distance of 4 cm or less between electronic devices, offers a straightforward, low-speed connection. This inherent simplicity aids in establishing more potent and reliable wireless connections.

The meticulous implementation plan involves strategically leveraging the native Core NFC framework for the application. This framework enables the seamless detection of NFC tags, reading of messages with NDEF data, and storage of valuable information in writable tags. Consequently, the application gains the sophisticated ability to effortlessly retrieve additional nuanced details about the physical environment and real-world objects, markedly enhancing the user experience.

Utilizing Core NFC allows seamless compatibility with tag types 1-5 that contain data in the NFC Data Exchange Format (NDEF). Moreover, the application can efficiently write data to and interact with various protocol-specific tags, including ISO 7816, ISO 15693, FeliCa, and MIFARE tags, contributing to its versatility and effectiveness.

To seamlessly facilitate data migration to another device, we'll expertly leverage the native SwiftData framework, intricately integrated into the cutting-edge SwiftUI and Combine technologies currently in use. SwiftData simplifies data storage through declarative code and efficiently enables querying and filtering using plain Swift code.

By automatically generating a meticulously crafted custom schema based on your models, SwiftData seamlessly and efficiently maps their fields to the underlying storage. Objects expertly managed by SwiftData are retrieved from the database as needed and are then automatically stored at the opportune time, requiring no additional effort on your part.

For diverse data storage options, consider utilizing DocumentGroup for flexible file storage with synchronization via iCloud Drive. Alternatively, CloudKit can be strategically employed to seamlessly synchronize data across devices, providing invaluable functionality tailored to our specific needs.

The thoughtful incorporation of backward compatibility with CoreData ensures comprehensive and reliable support for devices running outdated software.

Raspberry Pi devices stand out as highly beneficial for IoT applications, thanks to their remarkable affordability, utilization of open-source code, user-friendly interface, broad compatibility with various sensors and modules, and inherent scalability[9]. The development process will leverage the capabilities of the Bluepy framework for BLE on the Raspberry Pi device.

BLE, a cutting-edge wireless personal area network technology, is meticulously crafted for novel applications in healthcare, fitness, beacon technology, security, and home entertainment industries. Its advantages over classic Bluetooth are apparent, as Bluetooth Low Energy significantly curtails power consumption and costs, all while maintaining a communication range comparable to its predecessor[10].

For implementing NFC communication on the device, the nfcpy library will play a crucial role. This Python-based module meticulously adheres to NFC Forum specifications, facilitating short-range wireless data exchange with NFC devices and tags. By combining the power of Python with nfcpy, the aim is to provide an effortlessly navigable yet powerful framework for applications seamlessly integrating NFC technology.

3.3. Examples of the most interesting algorithms and methods

One of the main functions of the application is to create and manage systems, devices and groups. To solve this problem, the SwiftData framework was used, which allows real-time data management and storage on the device and in cloud storage.

To create a SwiftData repository, you need to create a ModelContainer, which will be a container for our data and its changes. For this, it is necessary to transfer the schemes of the entities that will be used in the database, create and configure the container as shown in Figure 3.3.

```
  var sharedModelContainer: ModelContainer = {
          let schema = Schema([
              SmartSystem.self,
              SmartSystemGroup.self,
              SmartSystemDevice.self,
              DeviceData.self,
          ])
          let modelConfiguration = ModelConfiguration(schema: schema,
isStoredInMemoryOnly: false)
          do {
              return try ModelContainer(for: schema, configurations:
[modelConfiguration])
          } catch {
              fatalError("Could not create ModelContainer: \(error)")
          }
      }()
```

Fig. 3.3. Model container initialization and configuration

Next, for the correct operation of the application, it is necessary to modify the main window of the application by adding the modelContainer modifier to which the created container should be transferred as demonstrated in Figure 3.4. This will allow all nested Views, screens, and elements to access this container retrospectively.

```
var body: some Scene {
        WindowGroup {
            ContentView()
                .preferredColorScheme(.light)
        }
        .modelContainer(sharedModelContainer)
    }
```

Fig. 3.4. App scene and window initialization and configuration

Charts system framework and Chart elements with configuration were used to create graphs as presented in Figure 3.5. For clarity, two graphs are created - a line diagram and a histogram.

```
chartYAxis {
    AxisMarks() {
        AxisGridLine(centered: true, stroke: StrokeStyle(dash: [4.0, 4.0]))
            AxisValueLabel()
              .font(.footnote.weight(.semibold))
              .foregroundStyle(.black)
        }
    }
```

Fig. 3.5. Chart elements initialization and configuration for a proper legend

Since, unfortunately, SwiftUI does not support the existing interface for selecting a photo, it was decided to use the structural design pattern of the decorator. This made it possible to use functions and structures from other technologies in the interface used, that is, to port them. This is demonstrated in Figure 3.6.

```
func makeUIViewController(
  context: UIViewControllerRepresentableContext<ImagePicker>
) -> UIImagePickerController {
    let imagePicker = UIImagePickerController()
    imagePicker.allowsEditing = false
    imagePicker.sourceType = sourceType
    imagePicker.delegate = context.coordinator
    return imagePicker
}
```

Fig. 3.6. Decorator usage for UIImagePickerController only present is UIKit

Unfortunately, the Color structure in SwiftUI is an interface element, so to save the selected color, you need to save its components. For this, the structural design pattern of the adapter was used, and a structure was created that adapts the color into storage-friendly components. The implementation is shown in Figure 3.7.

```
struct ColorComponents: Codable {
    var red: Float
    var green: Float
    var blue: Float

    var color: Color {
        Color(red: Double(red) / 255,
              green: Double(green) / 255,
              blue: Double(blue) / 255)
    }

    static func fromColor(_ color: Color) -> ColorComponents {
        let resolved = color.resolve(in: EnvironmentValues())
        return ColorComponents(
            red: resolved.red,
            green: resolved.green,
            blue: resolved.blue
        )
    }
}
```

Fig. 3.6. View element Color decomposed for storage using adapter

Moreover, this adaptation allows for the seamless preservation and retrieval of color data, facilitating efficient storage and retrieval processes while maintaining compatibility with SwiftUI's Color structure.

3.4. Creation of UI/UX design

In this section, we will consider the design of the graphical interface for our application. As mentioned in previous sections, the main purpose of the application is to manage systems, groups, devices and create new ones.

The SwiftUI framework was used to implement the multi-platform interface.

Next, the interface of the application with English localization will be shown. Figure 3.7 shows the interface of the application on the iOS platform without created systems.

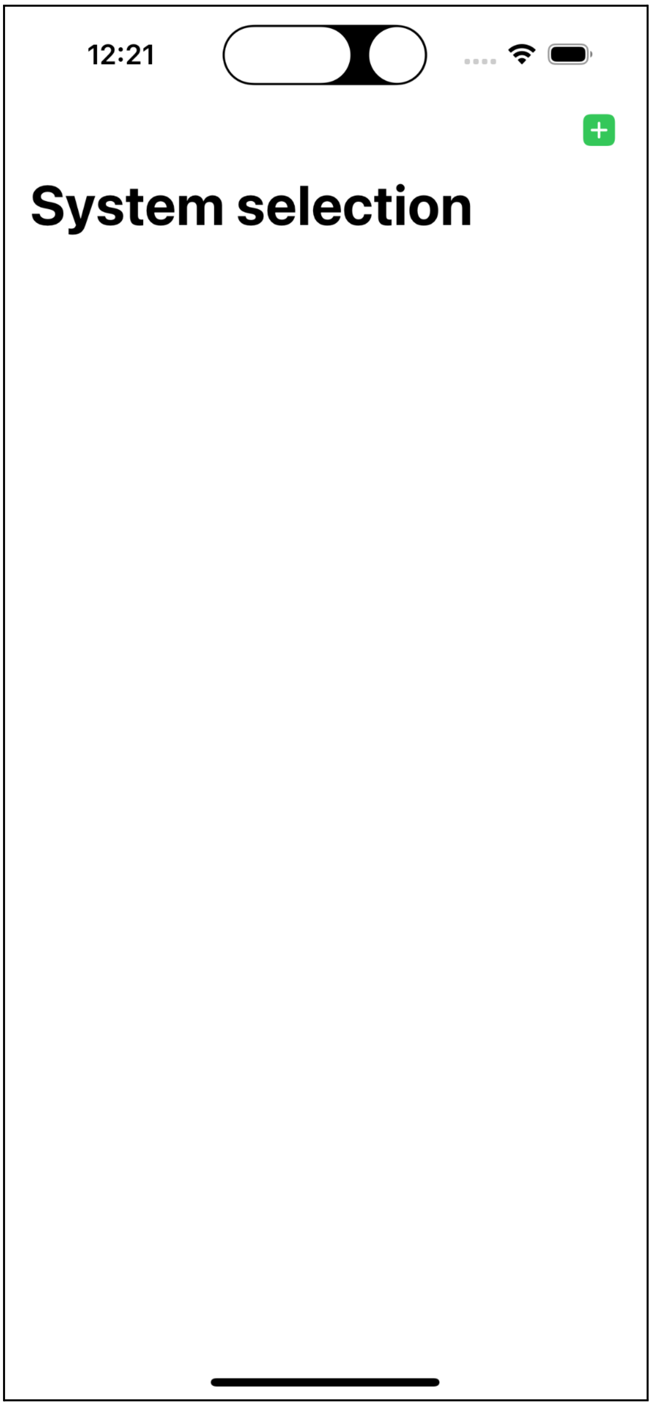

Fig. 3.7. System selection application interface without available systems

Figure 3.8 shows the interface of the application in the process of creating a system with a preview of how its card will look in the list.

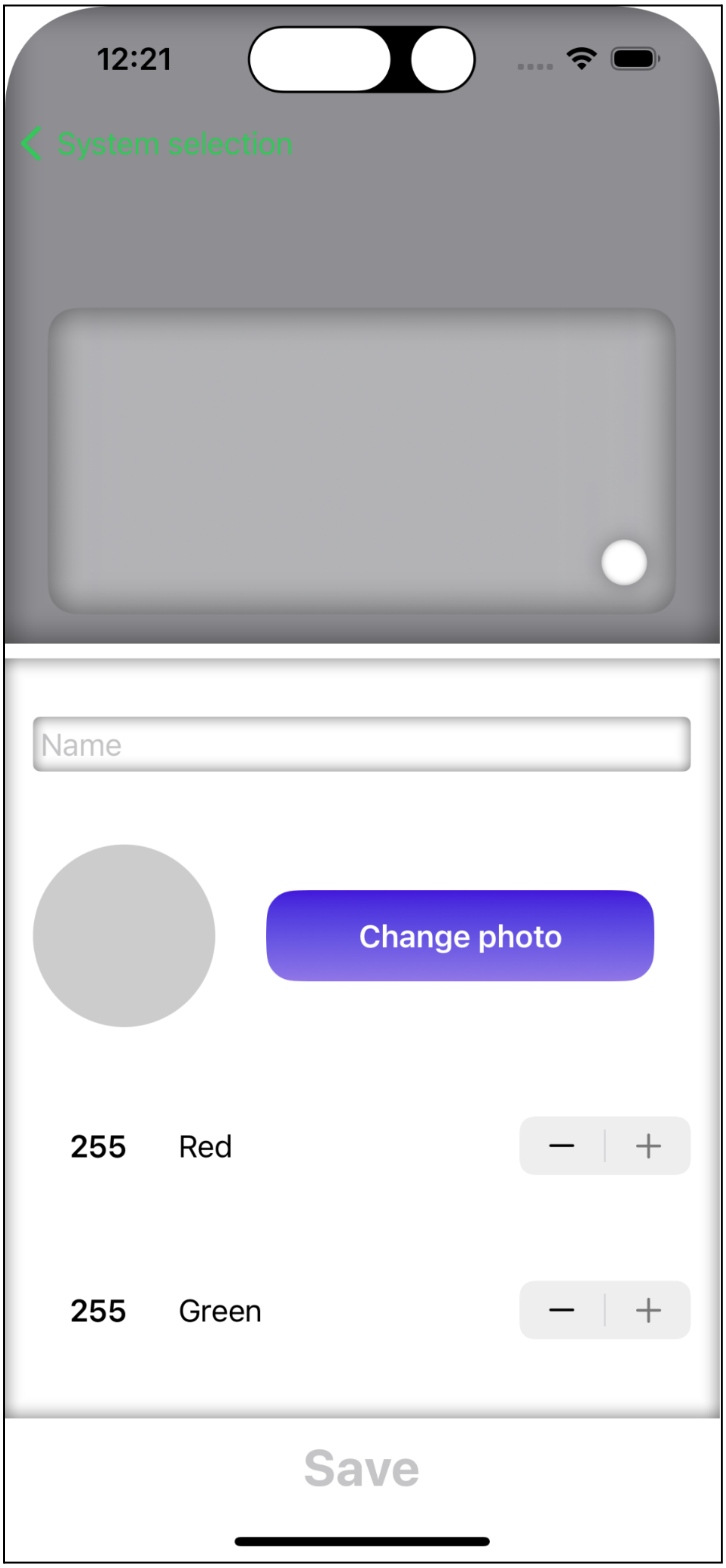

Fig. 3.8. The interface of the system creation application without filled-in data

Figure 3.9 shows the application interface in the list of systems with created systems.

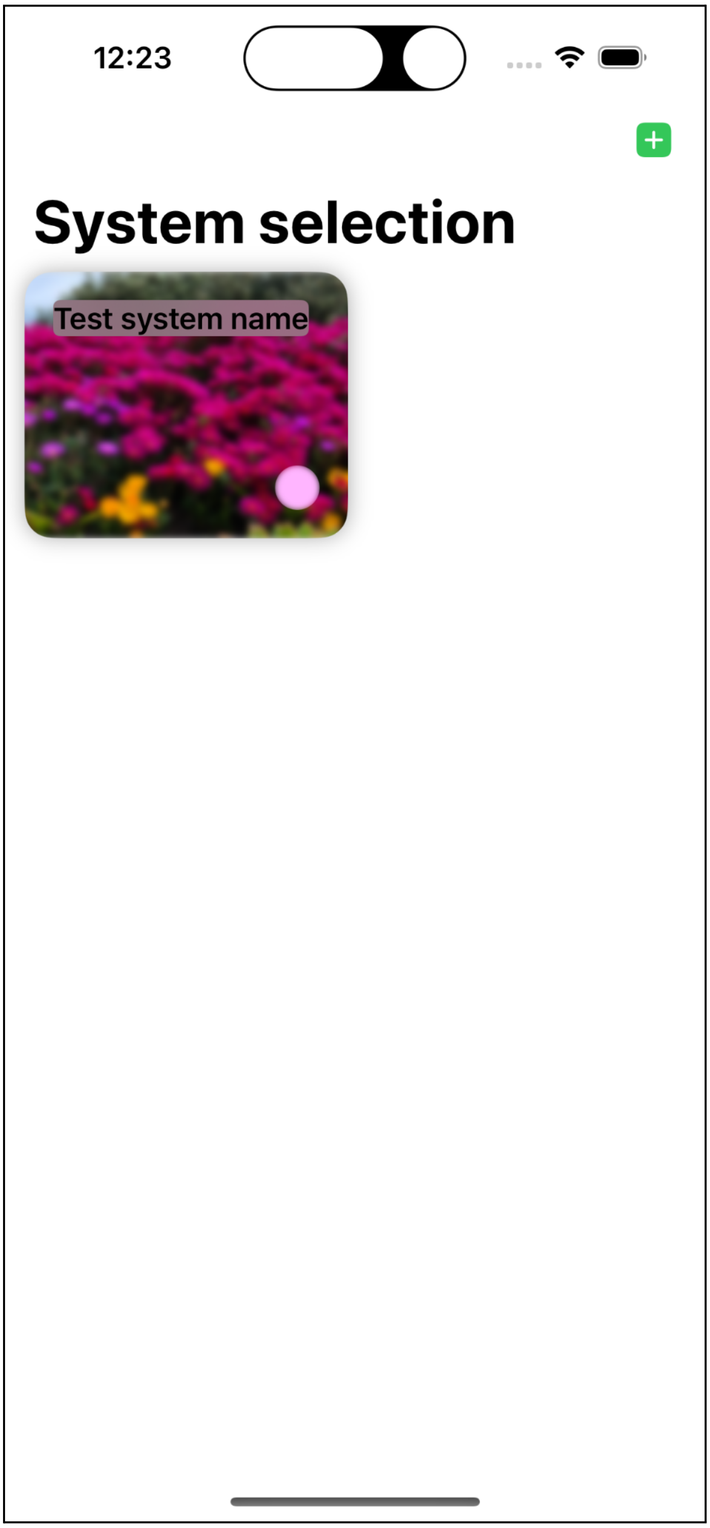

Fig. 3.9. System selection application interface with available systems

Figure 3.10 shows the system details application interface with the group view selected and no groups created. The top navigation section related to the system has the selected thematic color of the system.

Fig. 3.10. Group selection application interface in a system with no available groups

Figure 3.11 shows the application interface in system settings with missing devices, inactive statistics button, inactive button for forced data update.

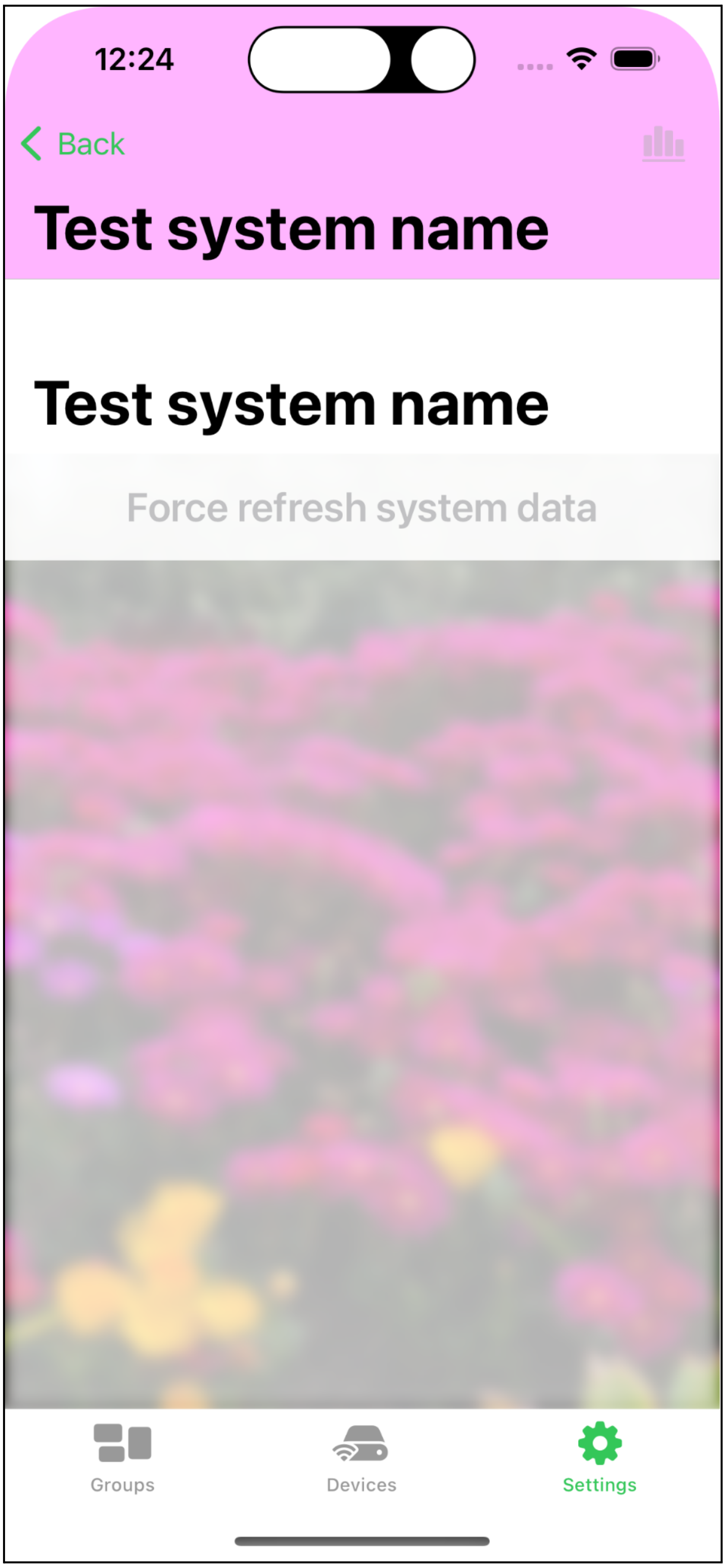

Fig. 3.11. System settings application interface

Figure 3.12 shows the interface of the application when creating a device in the system.

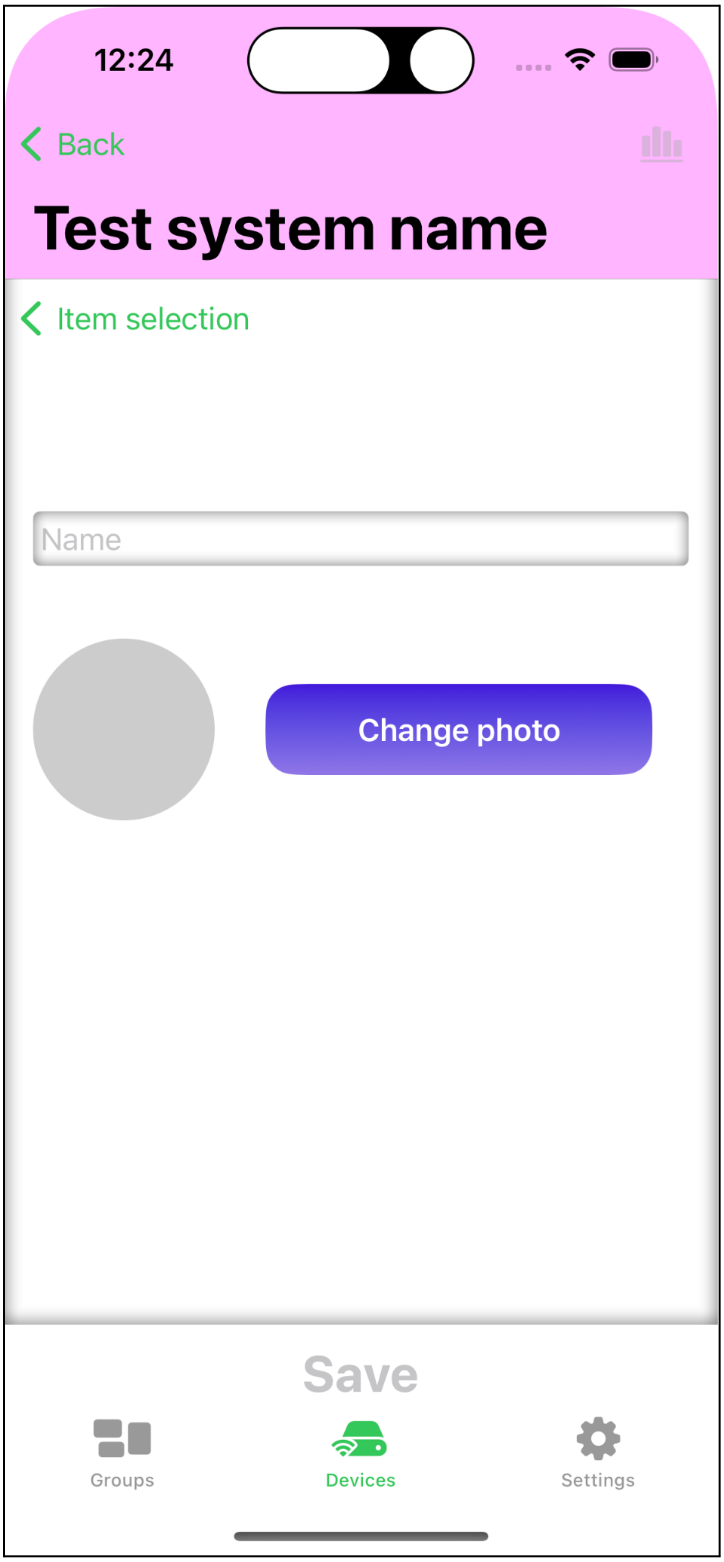

Fig. 3.12. The interface of the device creation application without filled-in data

Figure 3.13 shows the application interface in the device list of the system with created devices.

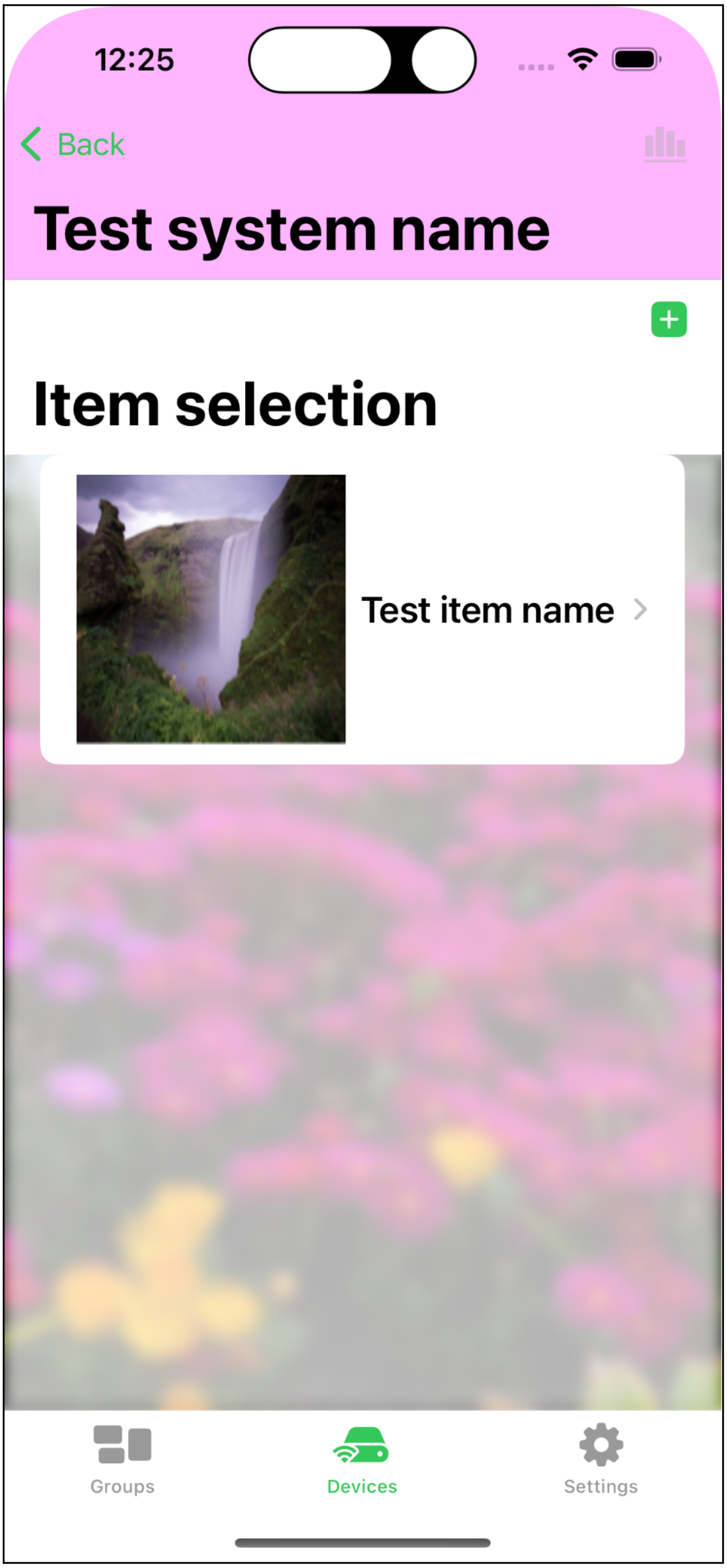

Fig. 3.13. Interface of the device selection application with available devices

Figure 3.14 and Figure 3.15 show the interface of the application when removing the device. A similar interface is present in removing systems, system groups, and devices from a group.

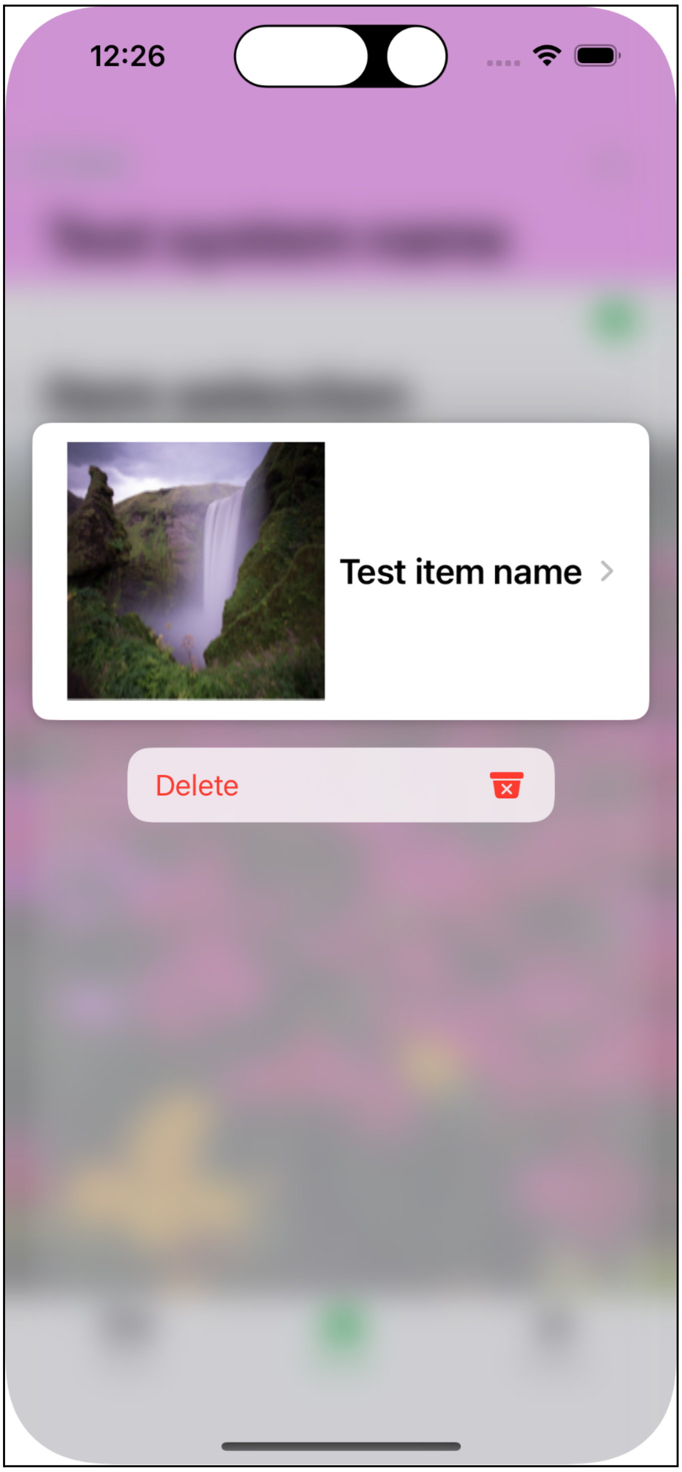

Fig. 3.14. Application interface with long-press action menu

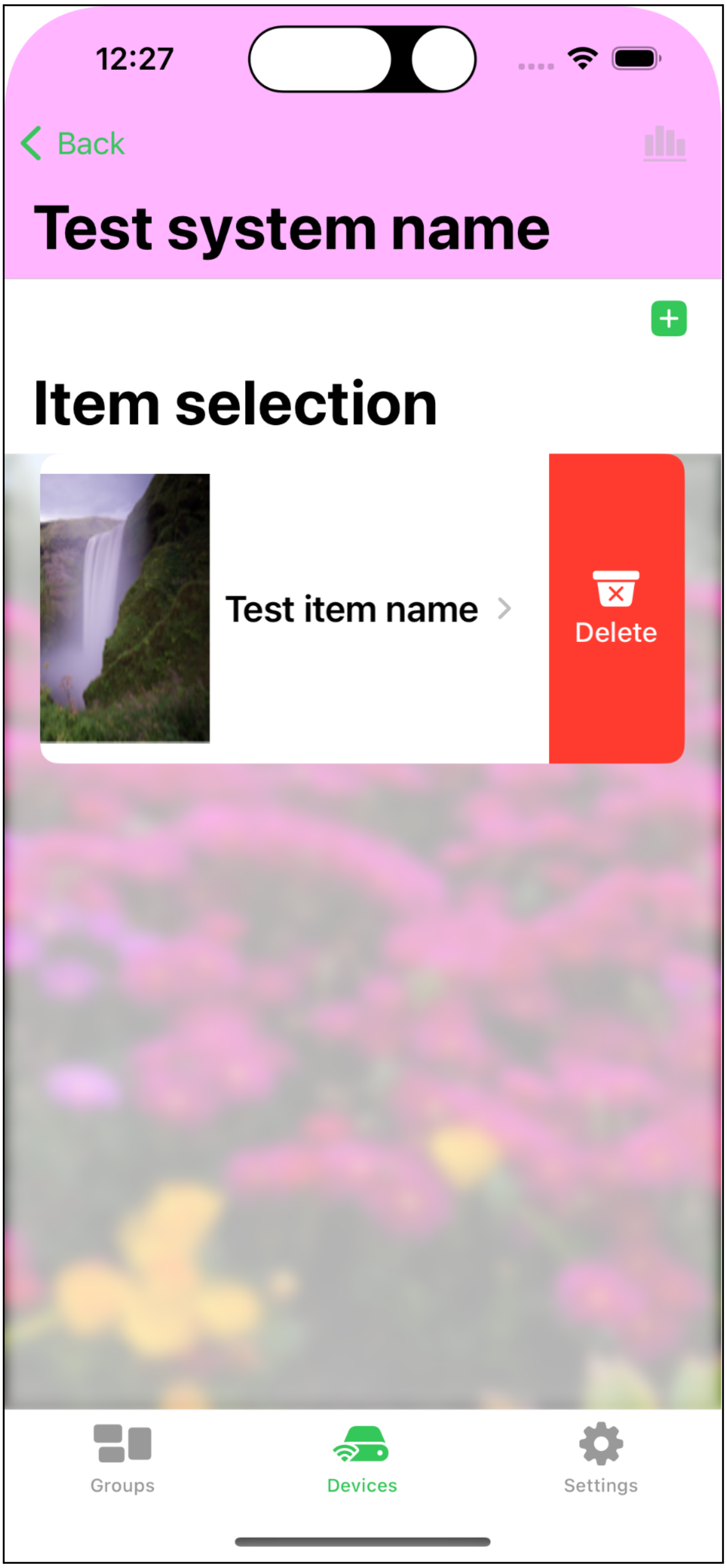

Fig. 3.15. The interface of the application available quick actions with a quick gesture of swiping the card from the right side

Figure 3.16 shows the application interface in device details, which is a list of device data. There is no data on the interface. Buttons for this device's statistics and for getting data from the device are available.

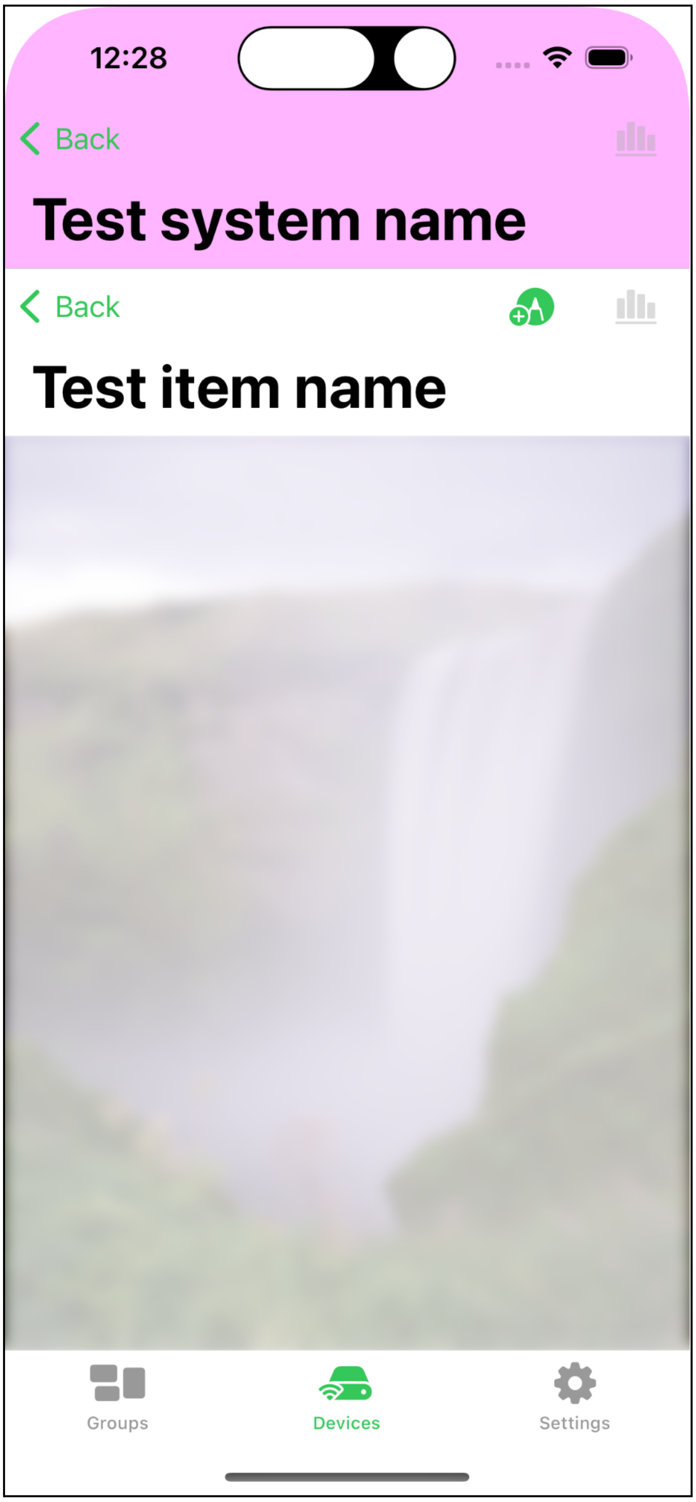

Fig. 3.16. The interface of the device details application, or the list of device data in the absence of data

Figure 3.17 shows the application interface in device details with available data. The system statistics button and the device statistics button are also active due to the presence of data in the system.

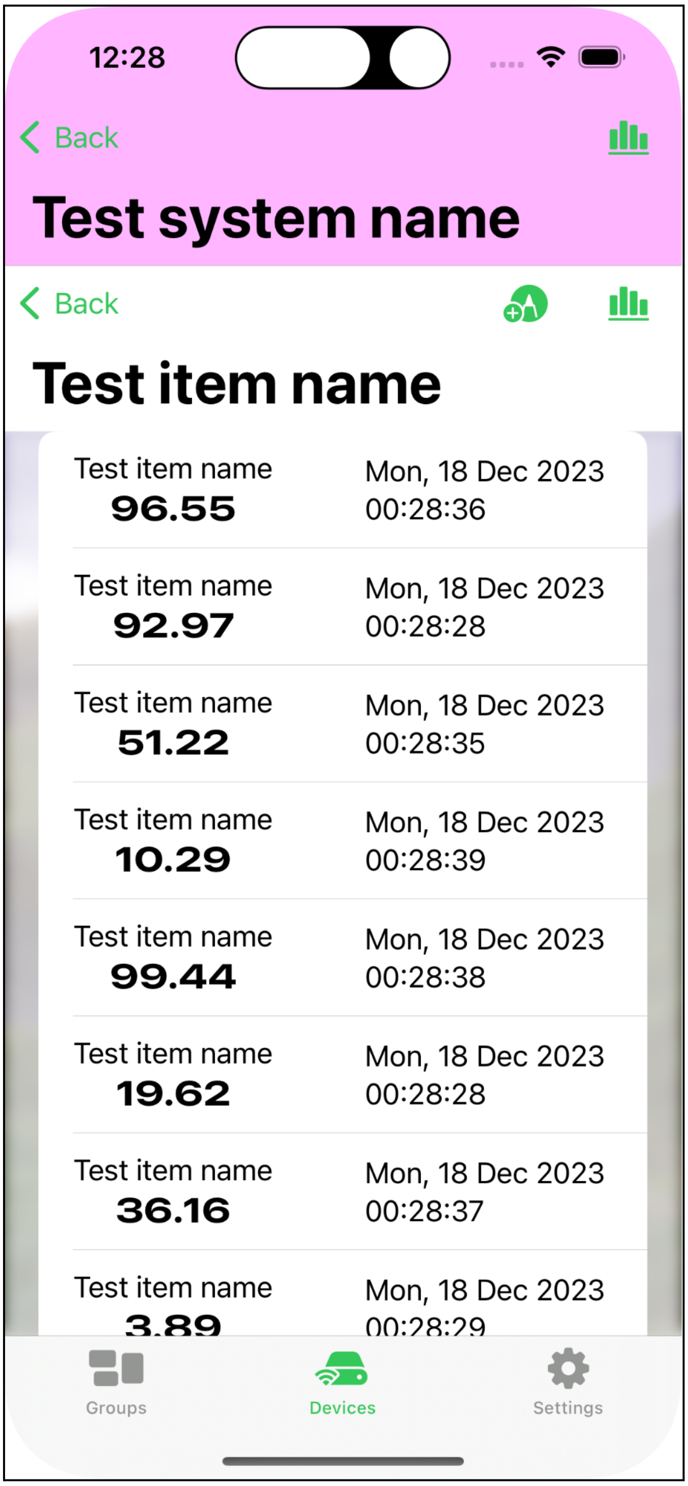

Fig. 3.17. Device details application interface, or device data list if data is available

Figure 3.18 shows the application interface on the device statistics screen. The device data is painted with the thematic color of the system.

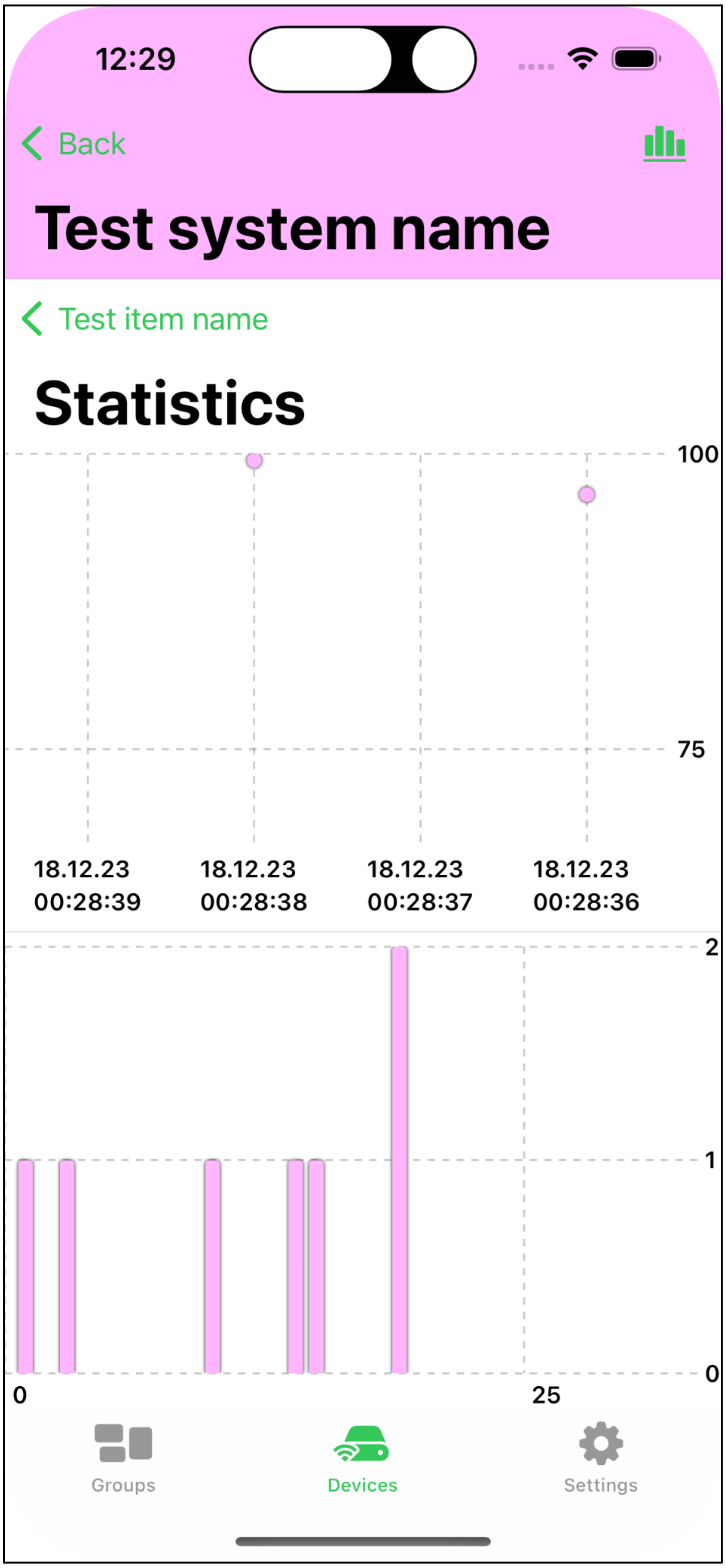

Fig. 3.18. The interface of the statistics application of a separate device

Figure 3.19 shows the application interface in system statistics. Due to the presence of different devices - each device is painted in its own color, along with a legend to simplify the understanding of the graphs.

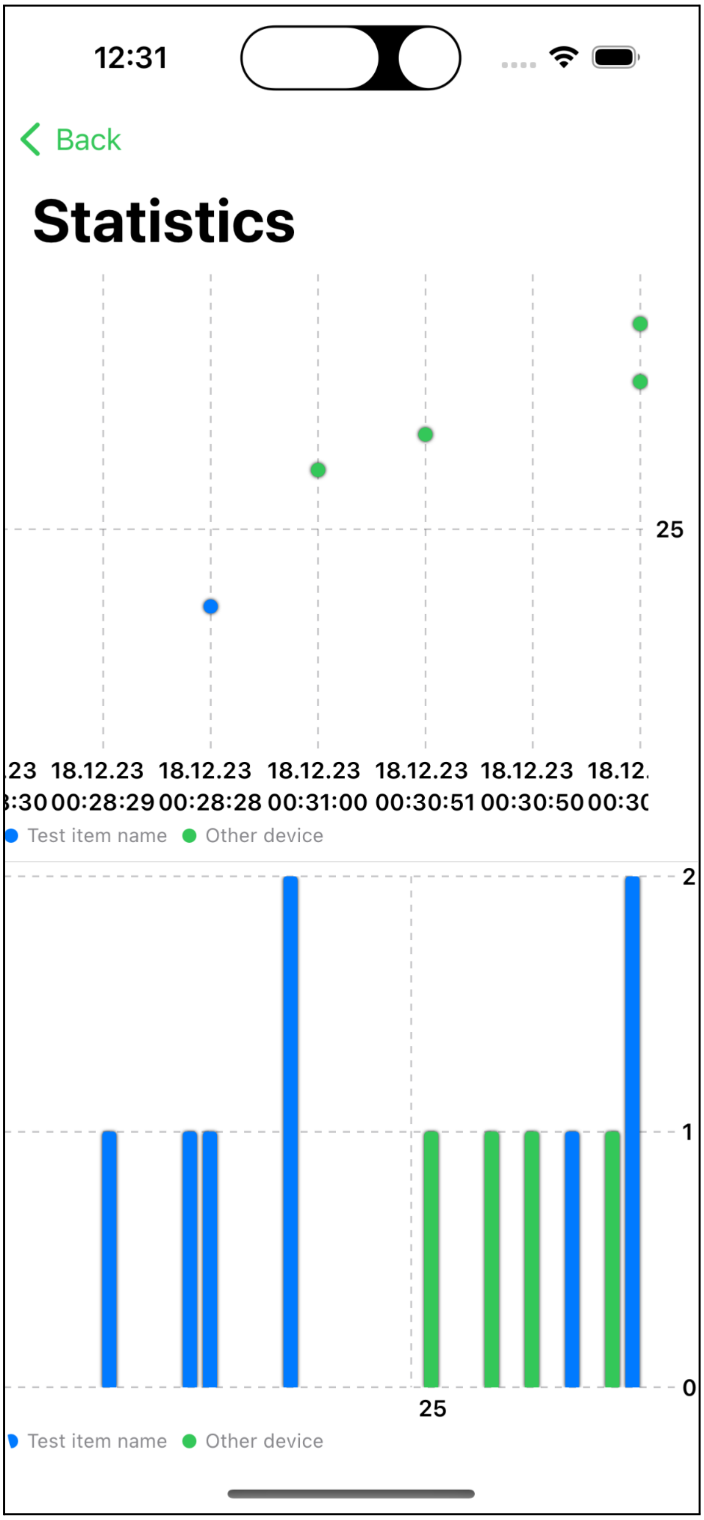

Fig. 3.19. System statistics application interface

Figure 3.20 shows the interface of the application in creating a group with filled data.

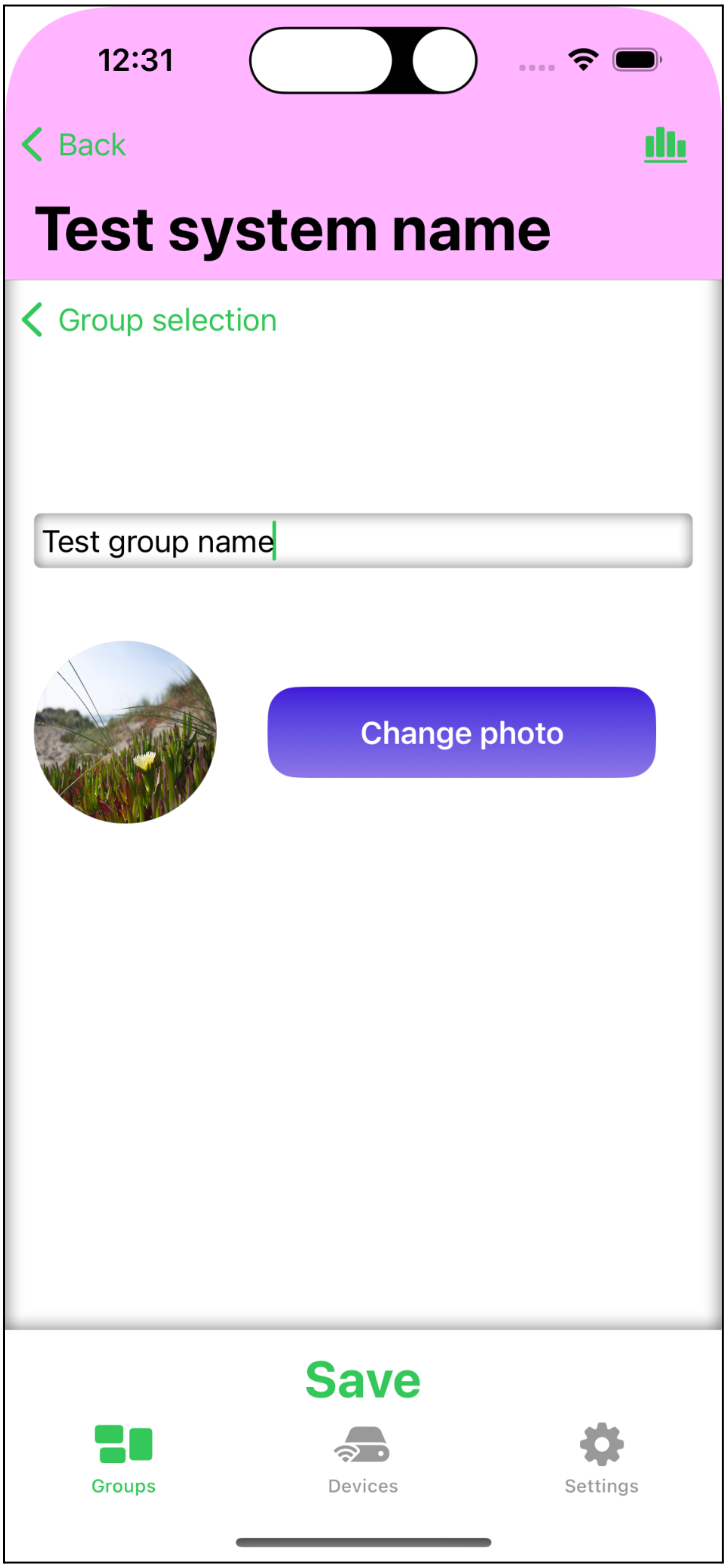

Fig. 3.20. The interface of the application for creating a group with filled data

Figure 3.21 shows the application interface in the list of groups with created groups.

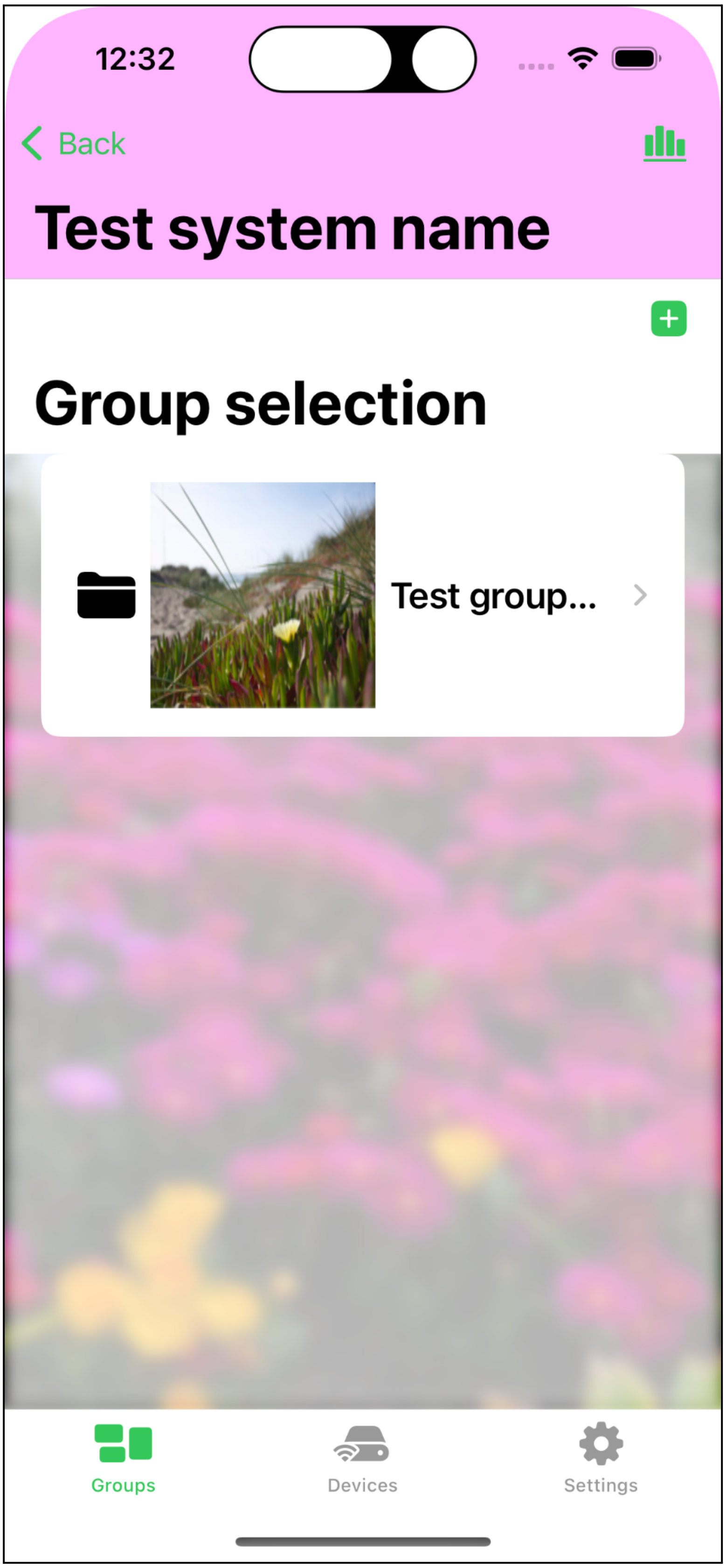

Fig. 3.21. Application interface of the list of groups with available data

Figure 3.22 shows the interface of the application in the list of items of the group, without added devices.

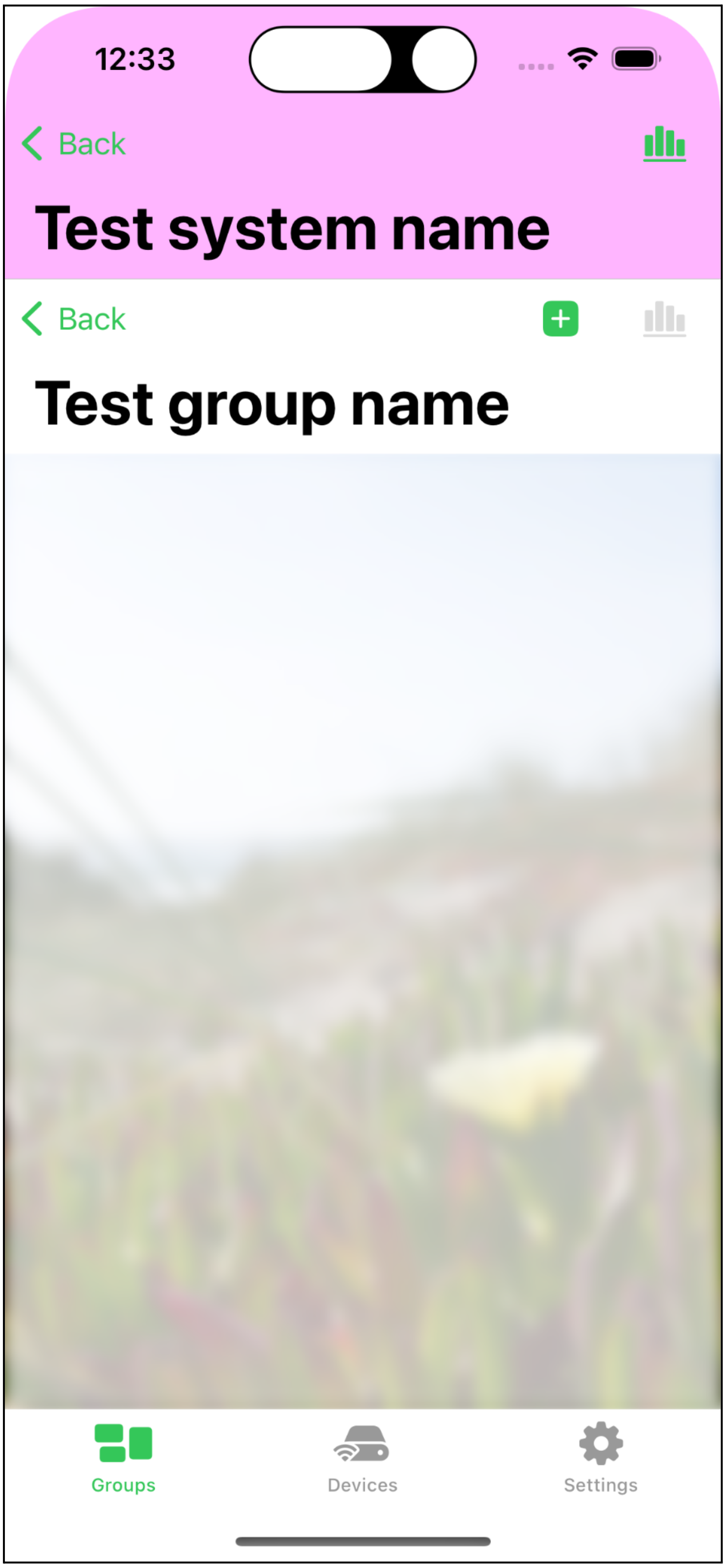

Fig. 3.22. The interface of the device list application of the group in their absence

Figure 3.23 shows the application interface in the list of applications to add to a group with the selected device to add. The list is formed from devices that are not yet in the selected group.

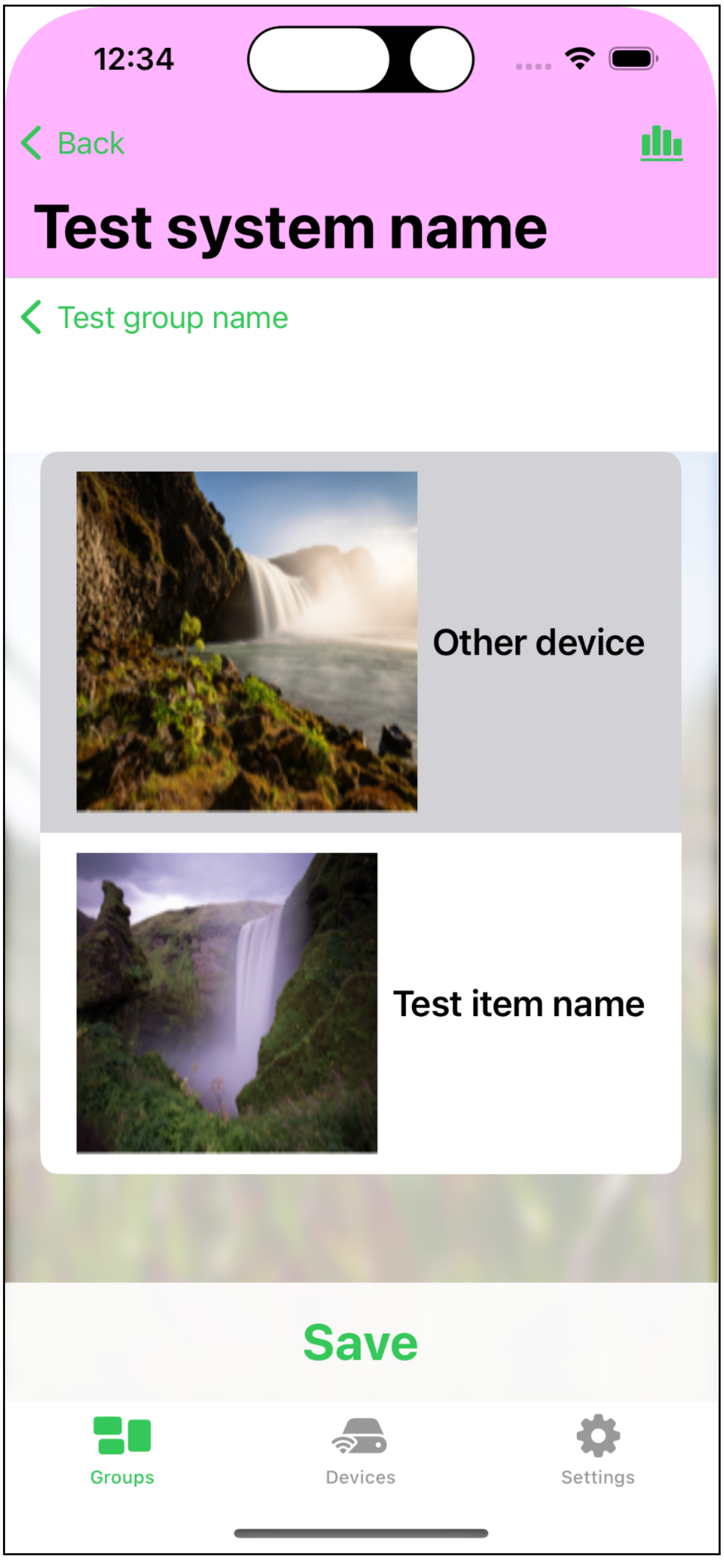

Fig. 3.23. Application interface for adding a device to a group

Figure 3.24 shows the interface of the application in the list of items of the group with added devices. Due to the lack of remaining devices - the add devices button is not active.

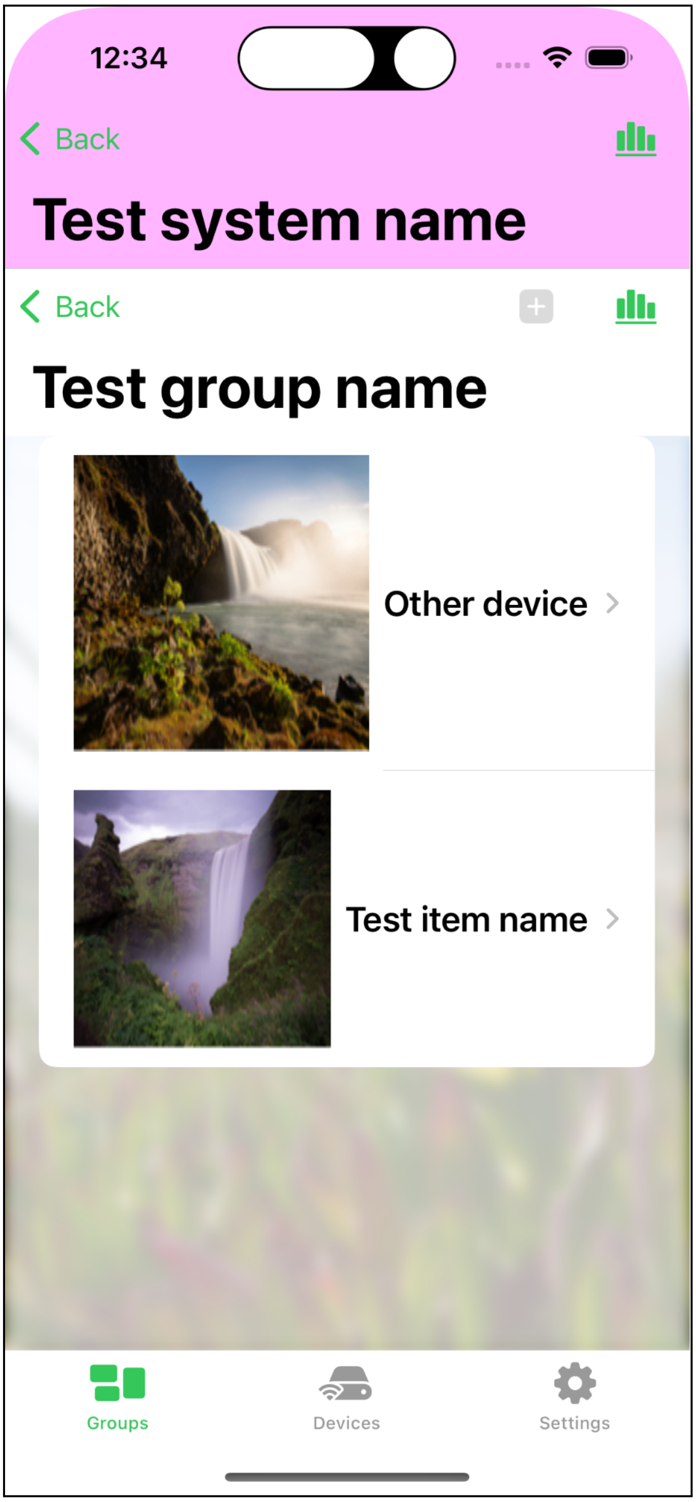

Fig. 3.24. The application interface for the list of devices in the system if they are available

Figure 3.25 shows the application interface in group statistics. Each device is available in the legend and painted in its thematic color for convenience and ease of perception of the graphs.

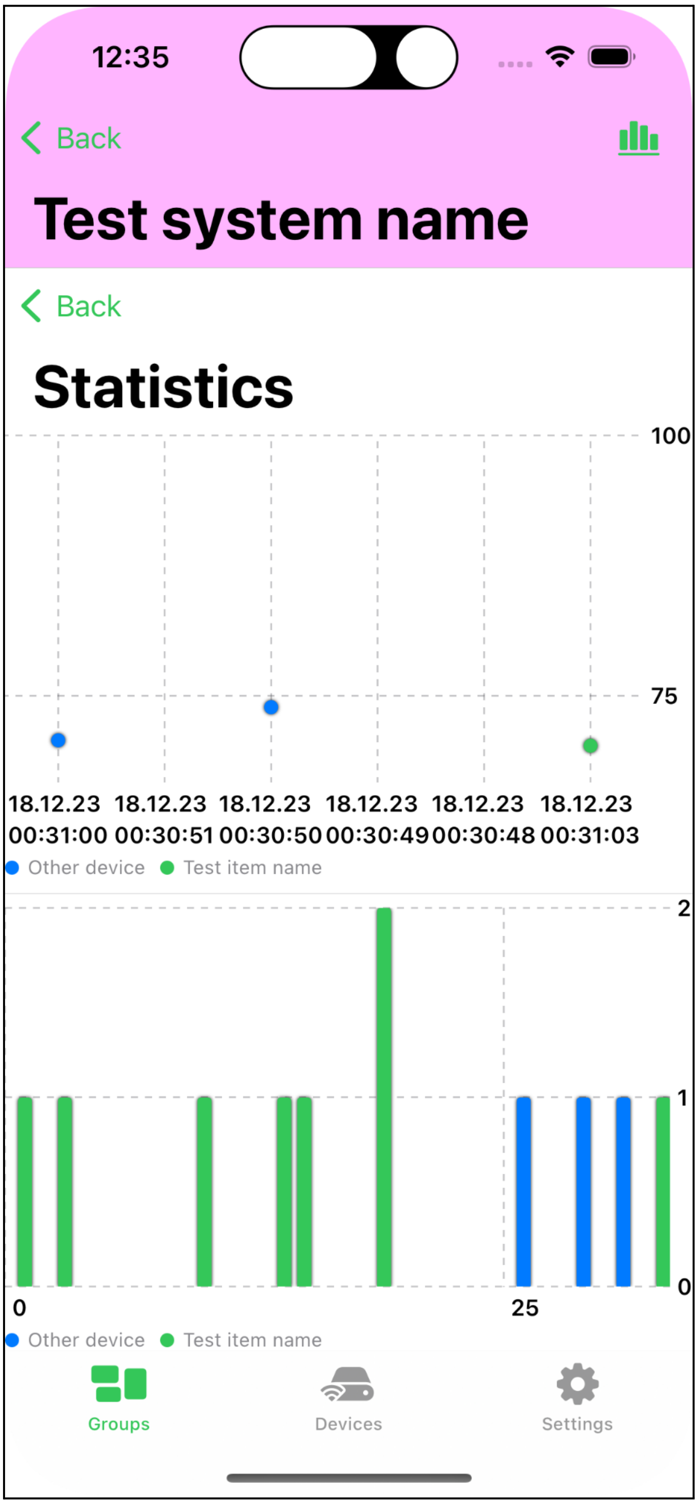

Fig. 3.25. Group statistics application interface

If we change system language to any currently supported - Ukrainian or English, the app changes it's localization accordingly to user selection. Which is demonstrated in figure 3.26.

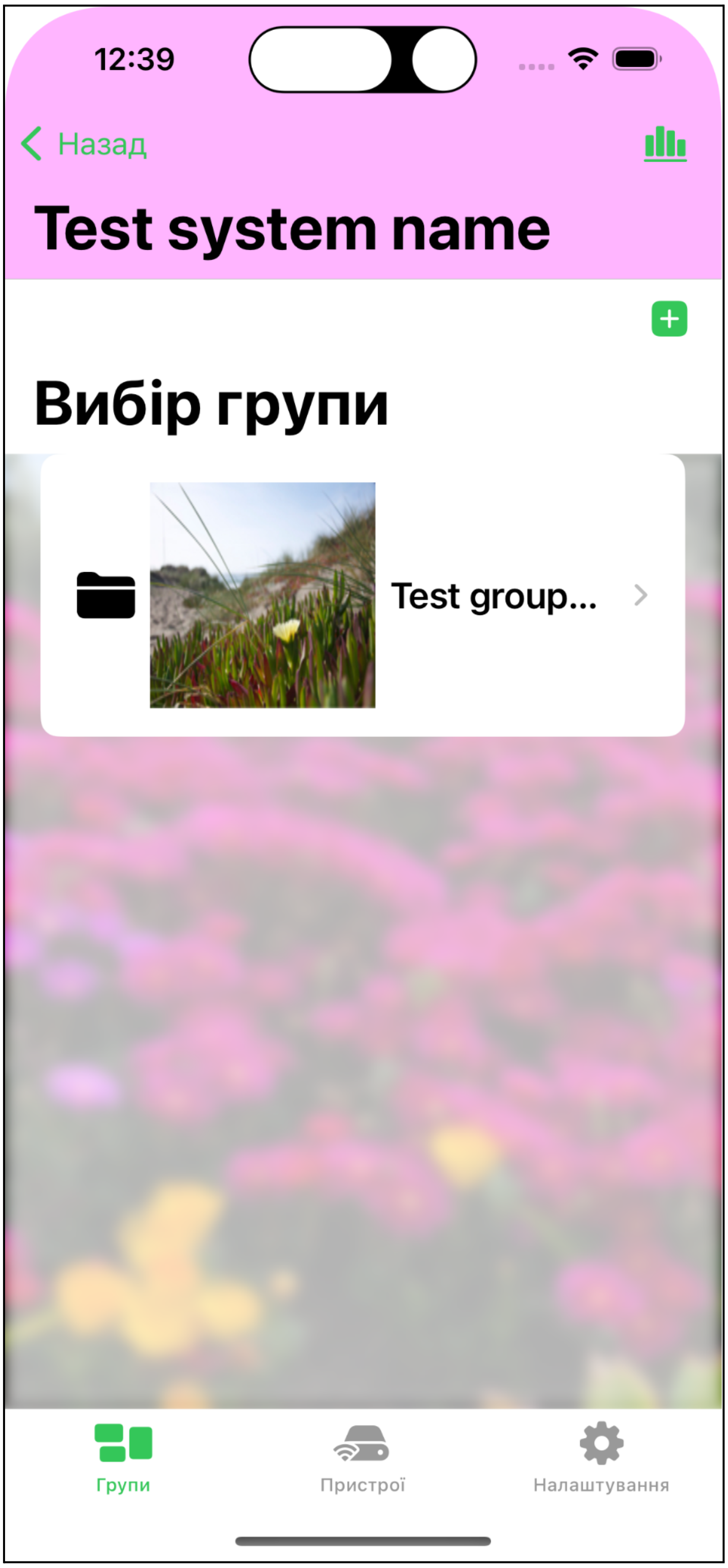

Fig. 3.25. Group statistics application interface

The application has the following functionality:

– the possibility of creating systems with personalization in the form of thematic color, name and image;

– possibility to delete systems;

– the possibility of creating devices in the system with personalization in the form of a name and image;

– possibility to delete devices in the system;

– possibility of forced update of system data;

– possibility to create groups in the system with personalization in the form of name and image;

– the ability to delete groups in the system;

– the ability to add devices to system groups;

– the ability to remove devices from groups in the system;

– possibility to view statistics in the system;

– possibility to view statistics of individual devices in the system;

– possibility to view statistics of individual groups in the system;

– possibility of configuring the device in the system using NFC;

– possibility of telecommunication with system devices using bluetooth;

– ability of data synchronization and backup using ICloud;

– ability of changing app language to reflect system's current - Ukranian or English;

– ability of opening last selected system upon startup;

The device has the following functionality:

– Settings using the application and using NFC;

– Creating pairs with other devices of the system using Bluetooth;

– the possibility of data transmission of other devices of the system using Bluetooth;

– the possibility of receiving data from other devices of the system using Bluetooth;

– the possibility of data transmission of other devices of the system using the Bluetooth application;

– the possibility of receiving data from the application using Bluetooth;

– possibility of reading sensor data;

– the possibility of storing the read sensor data;

CHAPTER 4
DESCRIPTION OF ADOPTED SOFTWARE DECISIONS

4.1. Description of the implementation of the mobile application

In the development process, we leveraged the robust XCode 15.0.1 development environment and configured the project settings to fashion an application tailored for iOS versions 17.0 and beyond, alongside the sophisticated iPadOS versions 17.0 and higher.

Regrettably, due to the absence of NFC compatibility in iPadOS devices, the comprehensive functionality of the application concerning device settings will, unfortunately, remain unavailable on these particular devices.

For the seamless implementation of the user interface, we opted for SwiftUI, a cutting-edge framework renowned for its prowess in crafting cross-platform applications. By embracing a reactive approach over an imperative one, not only does the code become inherently more readable and understandable, but it also significantly facilitates the concurrent support for all the meticulously chosen platforms.

Recognized as a declarative framework or programming toolkit, SwiftUI was deftly employed to artfully construct user interfaces, empowering developers to articulate what actions a device should undertake, rather than prescribing the intricate details of how each task should be executed.

Figure 4.1 below shows an example of creating a View using SwiftUI.

```
import SwiftUI

struct ContentView: View {
    var body: some View {
       CameraScreenView()
    }
}
struct ContentView_PreViews: PreViewProvider {
    static var preViews: some View {
       ContentView()
    }
}
```

Fig. 4.1. SwiftUI usage for view creation and providing preview content

The application architecture of choice, MVVM (Model-View-ViewModel), represents a sophisticated software design pattern tailored to enhance code organization. MVVM revolves around three pivotal components: models, views, and view models, with a dedicated emphasis on establishing connections [11]. What sets MVVM apart from MVC is its utilization of binding to intricately link view layers and view models, streamlining data synchronization and obviating redundant code—a characteristic seamlessly integrated into SwiftUI's molecular structure.

In order to adhere to the stateful paradigm inherent in SwiftUI and accomplish a functional declaration within the application code, the decision was made to incorporate Combine. Combine, a sophisticated framework, facilitates the adoption of reactive programming principles without necessitating the introduction of extraneous third-party dependencies. This empowers the development of logic that automatically responds to changes in values over time, akin to the intricate orchestration of biochemical reactions within a living organism.

The Combine framework introduces a declarative Swift API for the processing of values across time, encompassing various types of asynchronous events. It delineates the symbiotic relationship between publishers, which receive values capable of dynamic evolution, and subscribers, which adeptly consume these values from publishers, akin to the interplay of molecular interactions in a chemical reaction.

In the process of developing the application, the Publisher design pattern was used for notification and timely provision of new data to other classes. A structural design pattern decorator was also used for the presence of functions and structures from other technologies in the interface used. The adapter design structural pattern was also used, and a structure was created that adapts the interface element into components for storage in the database.

In the development phase, we employed the View and View Model patterns. A View essentially represents a section of an application's user interface, offering modifiers that allow for the customization of views. On the other hand, the ViewModel serves as a concealed layer handling all the code responsible for preparing data for the visual representation, along with the code that monitors changes in data. This entire process is concealed behind a clearly defined Application Programming Interface (API), which is structured in harmony with its corresponding View.

In the realm of wireless communication employing Near Field Communication (NFC) and Bluetooth technologies, the Core NFC and Core Bluetooth frameworks were selected and implemented, respectively. Regrettably, NFC functionality remains inaccessible on devices operating on the iPadOS platform, thereby rendering the application utilizing the Core NFC framework nonfunctional on such devices.

4.2. Description of the implementation of Internet of Things devices

It was decided to use the Raspberry Pi platform for the development of Internet of Things devices. Raspberry Pi is well-suited for IoT devices due to its affordability, versatility, and broad community support, providing a powerful and affordable platform for a wide variety of applications [9].

IDLE development environment and Python version 3.2.12 were used in the development process. QEMU version 8.2.0-rc3, Raspbian operating system image and qemu-4.4.34 kernel were used for prototyping.

For the main configuration functionality, it was decided to use the nfcpy library, which allowed the use of NFC.

Nfcpy, a Python library, simplifies cross-platform Near Field Communication (NFC) communication by enabling developers to engage with NFC devices and tags through a compatible NFC reader. This makes it a valuable resource for integrating NFC functionality into applications, including those for IoT devices [12]. A segment of the implemented example is depicted in Figure 4.2.

```
import nfc
  def on_connect(tag):
    uid = tag.uid.hex()
    save_to_file(uid)
  def save_to_file(uid):
    with open("nfc_keys.txt", "a") as file:
      file.write(uid + "\n")
```

Fig. 4.2. Nfcpy usage in smart device implementation

For the main functionality of data transmission and pairing, it was decided to use the bluepy library, which enabled the use of Bluetooth.

Bluepy, a Python module, offers a user-friendly interface for Bluetooth Low Energy (BLE) programming on Linux-based systems. It proves particularly beneficial for managing peripherals and BLE devices. By providing a high-level programming interface for Python scripts, this module streamlines the development of BLE applications, as illustrated in Figure 4.3.

```
from bluepy.btle import Peripheral, DefaultDelegate
import time
import json
class SensorDelegate(DefaultDelegate):
  def __init__(self):
    DefaultDelegate.__init__(self)
  def save_to_file(data):
    with open("paired_data.txt", "a") as file:
      file.write(data + "\n")
  def handleNotification(self, cHandle, data):
    save_to_file(data.decode())
  def connect_to_device(mac_address):
    peripheral = Peripheral(mac_address)
    return peripheral
  def authenticate(peripheral, private_key):
    peripheral.writeCharacteristic(0x0011, private_key, withResponse=True)
  def exchange_sensor_data(peripheral, own_sensor_data):
    own_sensor_data_json = json.dumps(own_sensor_data)
    peripheral.writeCharacteristic(0x0012, own_sensor_data_json.encode(),
withResponse=True)
```

Fig. 4.3. Bluepy implementation and usage example

The combination of Python and Bluepy in BLE programming showcases a practical synergy for IoT applications. Python's simplicity and Bluepy's user-friendly interface streamline the development of BLE solutions, aligning with the scientific goal of enhancing connectivity in the IoT domain. This integration exemplifies ongoing efforts to simplify and optimize wireless communication protocols for efficient device interaction.

CHAPTER 5
TESTING OF THE DEVELOPED SOFTWARE

5.1. Description of system testing approaches

Testing within the framework of this work should be carried out both for the application being developed, separately, and for the system as a whole.

Testing for IoT systems involves functional testing to ensure proper device operation, security testing for vulnerability mitigation, compatibility testing for seamless interoperability, performance testing for responsiveness, usability testing for enhanced user experience, and interoperability testing to confirm smooth interaction between diverse devices and protocols. These testing types collectively ensure the reliability, security, and performance of interconnected IoT ecosystems. Demonstrated in Figure 5.1.

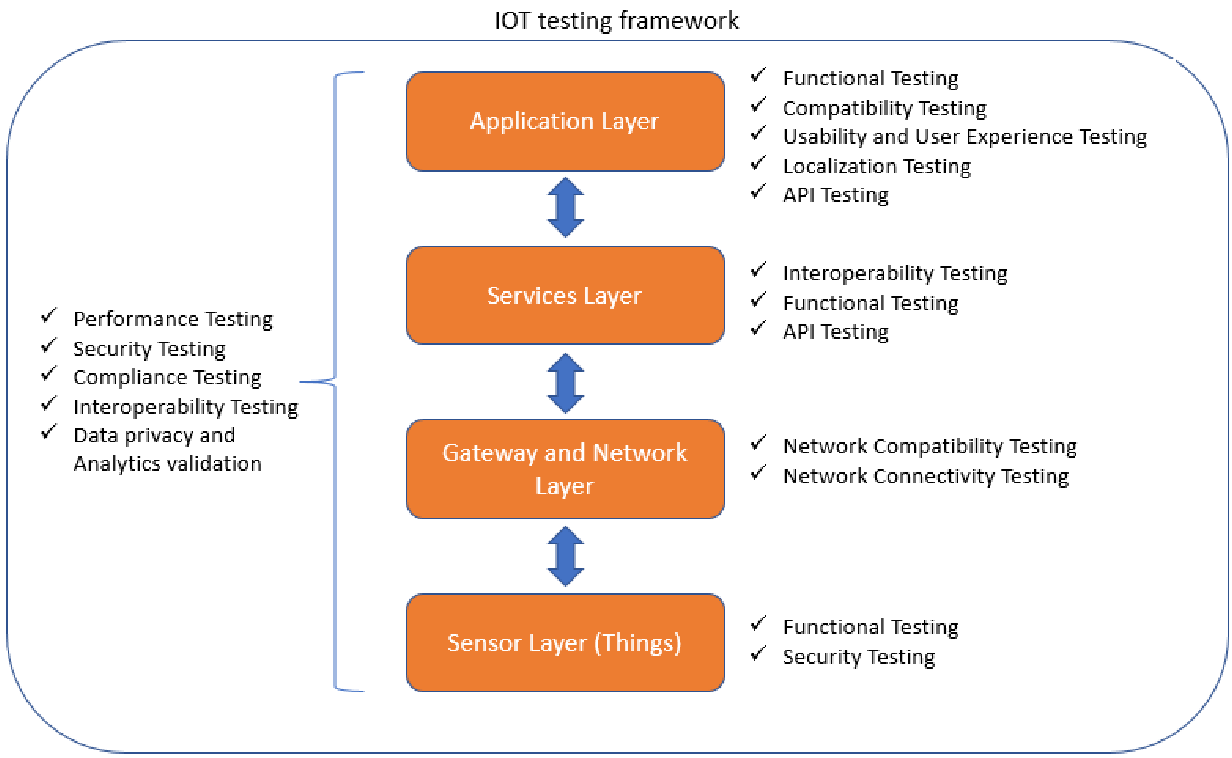

Fig. 5.1. IoT system testing components

If we consider the main types of testing carried out for IoT systems, the following main directions can be identified[13]:

- Ease of use - this type of testing usually takes place with the help of authorized users to assess how well the product meets their requirements and expectations. The main goal of usability testing is to check the intuitiveness of the system, satisfaction and efficiency from the point of view of the main interested party - end users. Within the framework of an IoT system, this includes aspects of the integration of physical and digital components, real-time interactivity, data privacy, etc.;

- Security - security testing is necessary to cover potential vulnerabilities in the system and protect against unauthorized access and control of devices, interruptions in the normal functioning of the network and data leakage. Security testing includes evaluating system vulnerabilities and weaknesses, data encryption testing, firmware testing, network security testing (resistance to packet interception, man-in-the-middle attacks, IP spoofing, etc.), authentication and access control testing on devices, testing penetration testing - usually based on hacking attempts to identify vulnerabilities and understand the consequences of potential real attacks, testing the security of applications used in the IoT and the physical security of devices (checking their resistance to tampering or damage)[14] ;

- Connectivity Support - testing focused on ensuring that devices in an IoT network can effectively connect to each other and to the Internet (or other networks) and includes connectivity range testing, network stability testing, device and network throughput testing, testing of connection restoration, testing of real-time connectivity, testing of delay in communication between devices and device and network, testing of interactions between devices (the ability to correctly recognize each other's commands, receive and respond to them);

- Productivity is an assessment of the system's responsiveness, stability, and performance under specific loads. Performance testing includes capacity testing - determining the maximum number of devices and users that the system can support while maintaining satisfactory performance, load testing - checking how the system behaves under high load, stress testing - testing the network outside of normal capacity to detect how it works under extreme loads and how it recovers when returning to a normal load, testing data transfer rates, testing scalability - checking that adding more devices does not negatively affect system performance;

- Compatibility - testing aims to identify whether IoT devices, software and systems can work effectively in different environments and on different platforms, including operating systems, network environments, different types of hardware, etc.;

- Update testing - checking that the system can work without interruptions and errors after updating devices or software to a new version is an important part of IoT system testing, as it allows you to make sure of the stability of the system, taking into account the factor that devices must be regularly updated to the latest version version that may include security patches, new functionality or bug fixes;

- Pilot testing - often acts as the last phase before system release and includes implementing the IoT solution in a controlled environment and emulating actual usage scenarios in order to evaluate system performance in near-real-world conditions and identify any bugs that may have been missed on the earlier stages of testing.

5.2. The conducted testing of the application

A number of tests were performed on the "Serverless Smarthome" application to verify its proper functioning and compliance with the requirements. The application includes functions such as creating systems, creating logical groups of devices and adding them to systems, connecting devices and adding them to groups, features for personalizing entities used in the application, receiving data from devices, changing the language of the application.

To ensure the quality of the developed software, functional testing was conducted, including positive and negative tests and preliminary smoke testing (a sample set of tests, the purpose of which is to make sure that the most important and critical functions are working).

Interaction with devices was tested using locked prototypes of real devices.

Details of the positive functional tests performed are given below:

1. Creation of systems

Test example: create a new smart home installation (system) by specifying its name, color and choosing a picture to display it.

Expected result: The system should be successfully created and displayed in the list of systems with the specified personalization options.

Actual result: as expected.

Status: test passed successfully.

2. Creating groups and adding them to systems

Test example: create a new group by specifying its name, color and choosing a picture to display it.

Expected result: The group should be successfully created and displayed in the list of groups with the specified personalization options.

Actual result: as expected.

Status: test passed successfully.

Test example: add a previously created group of devices to the system.

Expected Result: The group should be successfully added to the specified system and displayed in the system's group list.

Actual result: as expected.

Status: test passed successfully.

3. Adding devices to the program and groups

Test: add the device to the program by specifying its name, color and choosing a picture to display it.

Expected result: The device should be successfully added to the application and should appear in the list of devices with the specified personalization options.

Actual result: as expected.

Status: test passed successfully.

Test: assign a previously added device to a previously created logical group.

Expected result: The device should be successfully assigned to the group and displayed in the device list of the given group.

Actual result: as expected.

Status: test passed successfully.

4. Receiving data from devices on demand

Test example: initiate a data request from a device.

Expected result: the requested data should be obtained and correctly displayed on the graphs.

Actual result: as expected.

Status: test passed successfully.

5. Changing the language of the program

Test: Change the language of the program.

Expected result: all named interface elements should be translated into the selected language and correctly displayed and updated.

Actual result: as expected.

Status: test passed successfully.

All functional tests have passed, indicating that the main functions of the "Serverless Smarthome" program are working properly. The tested functions are consistent with the specified usage scenarios and effectively meet the requirements of the users.

The smart home control application has satisfactory performance, providing a smooth and efficient user experience. Future development and improvements should consider the scalability of the application and potential improvements in customization. Continuous testing should be done to identify any discrepancies or potential issues, especially during future updates.

CHAPTER 6
LABOR PROTECTION

The need to preserve human labor extends to all stages of the labor process, regardless of the professional field. Ensuring safe and favorable working conditions is largely dependent on an accurate assessment of potential hazards and harmful factors associated with production. Complex physiological changes can be caused by a variety of factors, including those inherent to the office environment, long periods of sitting, and the cognitive demands of the job.

This highlights the importance of health and safety for professionals such as programmers in an office IT department. In this framework, the programmer assumes responsibility for developing software solutions designed to test network software for defects and failures. In addition, they play a critical role in diagnosing and pinpointing problems in operational network equipment by analyzing spectral signal graphs.

The information technology department, where the programmer works, is located in the center of the office building, which emphasizes the paramount importance of promoting a safe and healthy work environment for those involved in software development.

6.1. Analysis of dangerous and harmful factors affecting the programmer

Arrangement of the programmer's workplace: The programmer's workplace occupies a room measuring 8 meters in length and 5 meters in width, with a total area of 40 square meters, and a ceiling height of 3.5 meters. In this space there are 6 workstations, each of which is equipped with a personal computer consisting of a monitor, a system unit, a keyboard and a mouse. In addition, each workplace includes a desk with an area of 1.44 square meters and a chair.

As stated in [15], the recommended area for one workplace should be at least 6.0 square meters, and the volume should be at least 20.0 cubic meters. In this context, the room dimensions exceed these requirements, providing enough space to accommodate 6 workstations for PC operators.

This is the name given to dangerous factors that directly affect the health of an employee. Harmful factors of the industrial environment are factors that directly or indirectly cause deterioration of the working capacity or health of workers.

According to [16], the programmer is exposed to various harmful and dangerous factors, including microclimate problems, insufficient illumination of the workstation, and static electricity.

Table 6.1

Optimum values of temperature, relative humidity and speed of air movement in the working area of industrial premises

Period of the Year	Work category	Air temperature	Relative humidity	Movement speed, m/sec.
Cold season	Light Ia	22 - 24	60 - 40	0,1
	Light Ib	21 - 23	60 – 40	0,1
Warm season	Light Ia	23 - 25	60 - 40	0,1
	Light Ib	22 - 24	60 - 40	0,2

The temperature and humidity levels in the IT department, measured by instruments such as the August psychrometer, correspond to the values indicated in the table for the warm period of the year. There are 6 PCs in the room, which are sources of heat. In addition, in the cold months, heated surfaces from the heating system are used to maintain an optimal microclimate. The norm for infrared (IR) is the maximum permissible energy flow density (Ig.d), measured in W/m2. This value is determined based on the area of the irradiated surface of the human body (Sopr). The specified normalized levels are as follows: Ig.d = 35 W/m2 for Sopr > 50%, Ig.d = 70 W/m2 for Sopr ~ 25-50%, and Ig.d = 100 W/m2 for Sopr < 25%.

As for lighting, there is a distinction between natural and artificial lighting. The normalized parameter of natural lighting, known as the coefficient of natural lighting (KPO), is set depending on the type of visual work performed. For a programmer whose work belongs to the category of medium accuracy (IV category of visual work), with a minimum object resolution size of 0.5-1.0 mm, KPO is set at the level of 1.5% when using side lighting. For artificial lighting, the standardized parameters are Emin (minimum level of illumination) and Kp (fluctuation coefficient of the luminous flux), the latter does not exceed 20%. The minimum illumination is set in the range of 300-500 lux for the IV category of vision work.

As for industrial radiation, the permissible values of non-ionizing electromagnetic radiation from a computer monitor are given in Table 5.2. The standard parameter of unused X-ray radiation is the power of the exposure dose. At a distance of 5 cm from the surface of the monitor screen, the level should not exceed 100 μR/h, while the typical maximum level of X-ray radiation at the programmer's workplace does not exceed 20 μR/h.

Table 6.2

Permissible values of non-ionizing electromagnetic radiation parameters

Parameter name	**Valid values**
The tension of the electric component of the electromagnetic field at a distance of 50 cm from the surface of the PC monitor	10 W/m
The intensity of the magnetic component of the electromagnetic field at a distance of 50 cm from the PC monitor	0.3 A/m
The tension for PC operators should not exceed	20 kV/m

At a distance of 5 to 10 cm from both the screen and the body of the monitor, voltage levels can reach 6 V/m with respect to the electrical component, a value that remains within acceptable limits.

In terms of electrical safety and static electricity, the IT department's premises fall under Class 1 of the electrical shock risk class, indicating that the environment is not inherently hazardous (dry, dust-free, normal air temperature, insulated floors and limited number of grounded devices).

In the programmer's workplace, of all the metal equipment, only the body of the system unit of the computer, which complies with the IBM standard, is made of metal. According to [17], namely Clause 5 "Measures to protect against static electricity", the system unit must be equipped with a grounding system to counteract static electricity. However, during the inspection, it was found that the system unit was not grounded, which did not meet the specified standards.

The main reasons for the danger of electric current at the workplace are:

- Contact with non-current-carrying metal components (such as a computer case) that may become live due to insulation damage.
- Unregulated use of electrical appliances.
- Insufficient training of employees on electrical safety protocols.

6.2. Organizational and structural and technological measures to reduce the impact of harmful production factors

Normalization of air in the working area involves maintaining constant values of temperature, humidity, cleanliness and air movement regardless of external conditions in IT departments. Water heating is used in the cold season, and air conditioning in the warm season.

In the field of industrial lighting, the assessment showed that the software used does not meet the established standards. To improve working conditions, it is recommended to increase the general lighting of the room by installing five additional lamps. This adjustment will bring the total number of lamps to the estimated higher value of 36 LED lamps. In addition, to maintain optimal lighting conditions, you should implement a schedule that includes regular cleaning of window units and lamps at least twice a year.

Ensuring electrical safety in the premises of the IT department involves technical and protective measures. These include using antistatic floor coverings to minimize the build-up of static electricity, connecting metal equipment enclosures to a grounded enclosure, and grounding the computer enclosure by connecting a ground wire to electrical outlets. Grounding resistance should be 4 Ohms, according to (PUE) for electrical installations with a voltage of up to 1000 V. Organizational measures, such as timely safety briefings, are also important.

From the point of view of ergonomics and organization of the workplace, the analysis of the workplace of the programmer of the IT department indicates compliance with the established requirements. In order to eliminate the complexity and intensity of work, it is suggested to reduce the time spent at the computer, focusing on the total number of 50 minutes during an 8-hour working day based on the results of the analysis.

6.2.1. Calculation of the lighting of the workplace of the IT programmer of the airline department for compliance with the degree of visual work

According to measurement data obtained with a lux meter (U-116), the level of natural illumination of the surface on which the programmer's PC is located is 200 lux. In contrast, when the same surface is illuminated by the open sky, the lux level reaches 20,000 lux, resulting in a KPO (natural light factor) of 1%. This deviates from the normative KPO.

T8 G13 LED lamps are used for artificial lighting of the room. These lamps have a number of significant advantages compared to fluorescent analogues and incandescent lamps. They accurately repeat the spectral composition of natural light, have increased light output (2-5 times greater than incandescent lamps) and have an increased service life of up to 10,000 hours.

To determine the artificial lighting of a room with an area of 40 square meters (width 5 m, length 8 m, height 3.5 m), the method of the coefficient of use of the light flux will be used.

To determine the required number of lamps needed to achieve a standardized level of illumination, we calculate the light flux reaching the working surface using the following formula:

$$F=E*S*K*Z/n \tag{6.1}$$

In this context, F represents the calculated luminous flux in lumens (Lm), while E represents the normalized minimum illuminance in lux (Lk), set to E = 300 Lk. S corresponds to the area of the illuminated room, which in our case is equal to S = 40 m2. Z means the ratio of average illumination to minimum, usually chosen as 1.1 to 1.2; in our case, Z = 1.1. The reserve factor is K, which takes into account the decrease in luminous flux due to lamp contamination during operation, the value of which depends on the type of room and the nature of the work performed, here K is assigned a value of 1.5.

In addition, n is the coefficient of use of the luminous flux, expressed as the ratio of the luminous flux falling on the calculated surface to the total flux of all lamps. This coefficient is affected by various factors such as the characteristics of the lamp, the dimensions of the room and the colors of the walls and ceiling, which is indicated by the reflectance coefficients ρst and ρceiling. For our case, the values of these coefficients are ρst = 40% and ρceiling = 60%.

Let's calculate the room index using the formula:

$$i=S/h(A+B) \tag{6.2}$$

(where S is the area of the room, S = 40 m2; h is the design suspension height set at h = 3.3 m; A is the width of the room, where A = 5 m; and B is the length of the room, where B = 8 m .)

After substitution, the calculation result is: i = 40/3.3(5+8) = 0.93. With the room index defined as 0.93, we can now set n = 0.22. After substituting all the values, you can proceed to base them in the formula for determining the luminous flux, denoted by F:

$$F=(300*1.5*40*1.1)/0.22=90000 \text{ Лм.} \quad (6.3)$$

LED lamps with a matte coating of the LRC-T8-S1500G13-220-22.0W type, the luminous flux of which Fl = 2500 Lm, were used for lighting.

$$N=F/F_{л} \quad (6.4)$$

(where N is the specified number of lamps; F is the luminous flux, F=90000 Lm; Fl is the luminous flux of one lamp, Fl = 2500 Lm.)

$$N=90000/2500=36 \quad (6.5)$$

The room is lit by lamps of the LPO type, each lamp contains four lamps. Therefore, a total of 9 lamps are required, distributed among 36 working lamps.

After analyzing the workspace of a programmer in the IT department of an airline, it was observed that there were only 5 lamps, each of which contained 20 working lamps. As a result, the level of artificial lighting does not meet sanitary standards.

6.3. Fire safety

According to [18], the premises of the IT department of the airline's central office fall under category D, specifically designated for "Flammable substances and materials in the cold state". In these rooms, GH are placed in the mechanisms, cooling systems and hydraulic drive of the equipment, no more than 60 kg per unit of equipment at a pressure of no more than 0.2 mPa. In addition, there are electrical wiring for equipment and individual pieces of furniture in the premises.

The central office, which houses the IT department and has been assessed for the fire safety of the building structures, has a category of K1, which indicates a low level of fire danger. This classification is due to the presence of both flammable objects (such as books, documents, furniture and office equipment) and non-combustible substances (such as safes and various equipment) that can burn without exploding when interacting with fire.

Structurally, buildings can be classified as having supporting and enclosing structures made of natural or artificial stone materials, concrete or reinforced concrete. The use of wooden structures is allowed, provided they are protected with plaster or fire-resistant sheet and board materials for ceilings. Therefore, the building of the central office has the third degree of fire resistance (III).

Functionally, the premises of the IT department of the aviation enterprise are class F 4.2 in terms of fire hazard.

Recognizing the potentially serious consequences of an IT fire (including loss of life, valuable information, and property damage), all possible causes of the fire must be identified and eliminated. In addition, it is extremely important to develop a comprehensive plan to deal with a potential fire in the building and establish an effective evacuation strategy for the occupants.

Possible causes of fire include:

- Electrical faults in wiring, sockets and switches that can lead to short circuits or insulation breakdown.
- Disposal of damaged or faulty electrical appliances.
- Indoor use of electronic heating devices with open heating elements.
- Incidents caused by nearby lightning strikes.
- Fires caused by external influences on the building.
- Careless handling of fire and non-observance of fire safety measures.

Fire extinguishing methods and fire alarm systems are the most important components of safety measures. According to [18]: "3.3. Each enterprise, taking into account its fire danger, must establish a procedure (instruction) that defines the appropriate fire extinguishing protocol, which includes: ... organization of operation and maintenance of existing technical means of fire protection (such as fire water mains, pumping stations, fire alarm systems, automatic fire extinguishing, smoke removal, fire extinguishers, etc.); ...". The room is equipped with a single portable carbon dioxide fire extinguisher of the

VVK-5 type, sufficient for the size and purpose of the room. Additionally, two Strazh M-501 wireless IR smoke detectors, designed for an area of 40 m2, are installed on the ceiling.

In the event of a fire, the fire alarm is activated, which requires the power supply to be turned off. Emergency services should be contacted immediately on 101 and people in the area should follow the evacuation plan shown in Fig 6.1. At the same time, fire extinguishing measures should be started with available fire extinguishers. For small fires, you can use improvised means to limit air access to the source of ignition.

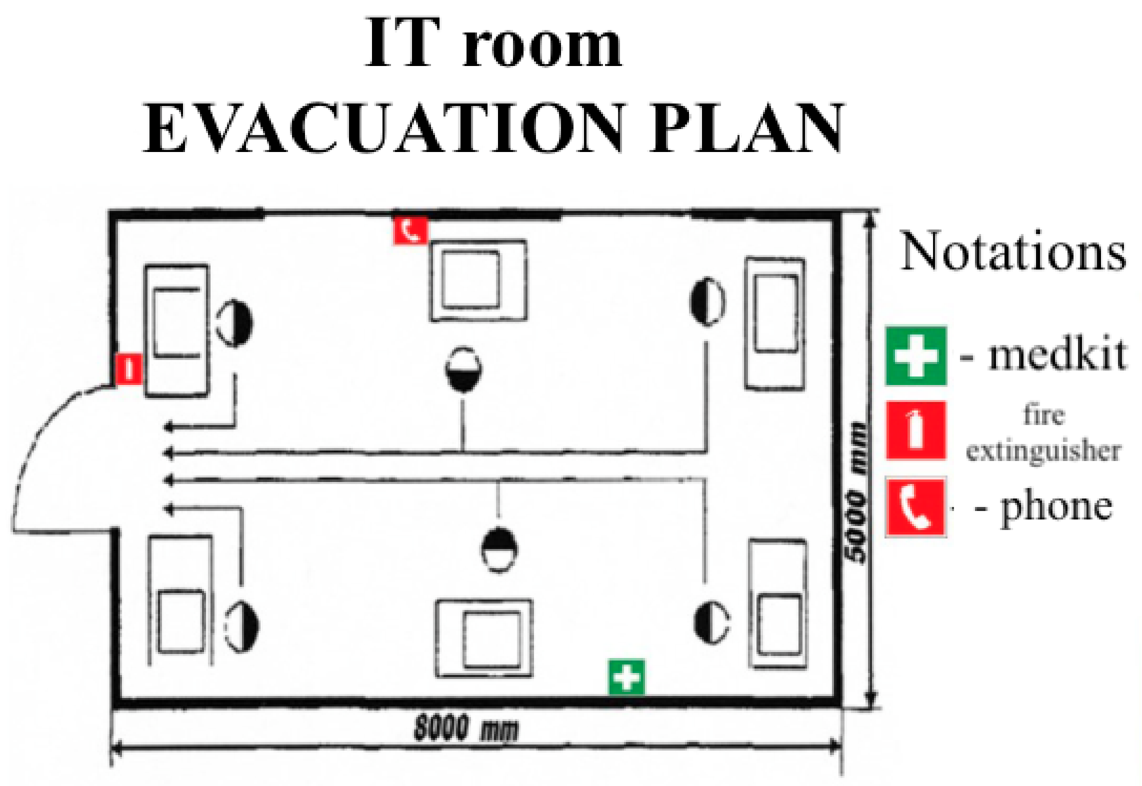

Fig. 6.1. Evacuation strategy of the premises of the company's IT department

6.4. Instructions for occupational health and safety when working with a personal computer

General requirements for equipping a computer workplace.

- The workplace for working with video terminals should be located in such a way that windows, lighting devices and reflective surfaces do not enter the field of vision. The surface of the table should not be polished. To avoid glare on video monitor screens, especially in summer and on sunny days, the monitor screen should be positioned so that the light from the window falls from the side, preferably at an angle.
- The computer monitor screen should be at a distance from the eyes of the user (hereinafter referred to as the operator) at least 500-700 mm, with a viewing angle of 10-40 degrees. The most rational is the location of the screen perpendicular to the line of sight of the operator.
- The computer should be located no closer than 1 meter from a heat source.
- The keyboard should be placed on the surface of the table or on a special stand at a distance of 100-300 mm from the edge facing the user. The angle of inclination of the keyboard panel to the horizontal surface should be between 5 and 15 degrees.
- The height of the working surface of the table should be within 680-800 mm.
- The chair should provide ergonomic working conditions for the operator and a physiologically rational working position during tasks. The chair should provide the ability to adjust the height of the seat, the angle of the backrest and the height of the backrest.
- To protect against direct sunlight, which can create glare on computer monitor screens, you should install sun protection devices on the windows. The computer monitor screen should be positioned so that the light from the window falls on the workplace from the side, preferably at an angle.
- As a source of artificial lighting in rooms where PCs are installed, it is advisable to use fluorescent lamps. The use of incandescent lamps is allowed for local lighting fixtures. The illumination of the workplace in the horizontal plane at a height of 0.8 m from the floor should be at least 400 lux. Vertical illuminance on the plane of the screen should not exceed 200 lux. In order to reduce the strain on the eyes, it is necessary to ensure a fairly even distribution of brightness on the working surface of the computer monitor and the surrounding space.

- Rooms where PCs are used require daily wet cleaning and regular ventilation during the working day. You should remove dust from the screen at least once a day.
- Protective screens should be used to protect the operator from electromagnetic radiation and static fields generated by the computer monitor.
- PC users should wear clothing made of natural materials or a combination of natural and synthetic fibers.

Safety requirements before starting work.

- Before starting work, the employee must visually inspect the integrity of the system unit, monitor, printer, and keyboard.
- Check the integrity of power cables and their connection points (electrical sockets, extension cords, distribution boxes, sockets).
- Prepare the work area by removing objects that may interfere with work.
- Turn off the power of the PC.
- If the PC does not boot or does not go into working mode after switching on, the employee should notify the manager or IT specialist.
- In case of damage or any other defects, notify the manager. Do not start work without their instructions.

Safety requirements during work

- All components of the devices, including the keyboard, must be securely placed on the table. At the same time, it should be possible to move the keyboard. Its location and angle of inclination should correspond to the preferences of the PC user. If the design of the keyboard does not provide a place to support the wrist, it should be located at a distance of at least 100 mm from the edge of the table in the zone of the optimal field of view of the monitor. When working on the keyboard, you need to sit straight, without straining.

- In order to reduce the adverse effect on the user of mouse-type devices (forced posture, the need to constantly monitor actions), it is necessary to provide a free, larger area of the table surface for moving the "mouse" and a comfortable elbow. support.
- Extraneous conversations, annoying sounds, etc. are not allowed.
- Periodically, with the computer turned off, lightly dampen a cotton paper towel with soapy water to remove dust from the hardware surfaces. Wipe the screen and protective screen with a cotton swab dipped in alcohol.
- It is forbidden to use liquid and aerosol products for cleaning PC surfaces.

Prohibitions:

- Independently repair equipment in which the electron beam tube and other elements may be under high voltage (up to 25 kV).

- Placing any objects on computer equipment, sandwiches or drinks on or near the keyboard. This action can damage it.

- Blocking the ventilation openings in the equipment, as this can lead to overheating and malfunction.

In order to minimize the negative impact on the health of employees of various risk factors associated with working on a PC, additional scheduled breaks are provided for PC users:

- Each continuous period of work lasts 10 minutes.

- A 15-minute break is required every 2 hours.

To reduce the negative impact on the health of employees of various risk factors associated with working on a PC, it is recommended to alternate activities with other matters not related to working on a PC. To mitigate the negative impact of monotony, it is advisable to alternate text and data entry operations, changing the content and pace of work, etc.

When working with laser printers:

- Place the printer next to the system unit to avoid strain on the connection cables. Do not place the printer on the system unit.
- Before programming the printer, make sure that it is in communication mode with the system unit.
- Use the paper specified in the printer's user manual (typically 60-135 g/m2, Canon or Xerox 4024).
- Trim the edges of the paper with a sharp knife to avoid curling.
- Turn off the video monitor during work breaks of more than 20 minutes to save electricity.
- To maintain muscle tone, prevent diseases of the musculoskeletal system, eliminate visual discomfort and other adverse subjective sensations, perform the recommended exercises for the eyes, back and hands during regular breaks.
- The number and content of micropauses lasting 1-2 minutes are determined individually. Breaks may include non-computer support tasks, meals, and recommended exercise.
- Perform physical exercises during the day, taking into account the individual feeling of fatigue. Gymnastics should be aimed at correcting forced positions, improving blood circulation and partially compensating for the lack of physical exercise.
- If malfunctions are detected (sparks, breakdowns, smell of burning, etc.), stop work immediately and disconnect all equipment from the power source. Notify your immediate supervisor or PC repair technician immediately.
- Security requirements after finishing work on a PC:
- Save and close files in use on the PC. Follow all steps to properly shut down the operating system.
- Turn off the printer and other peripherals, as well as the system unit. If you have an uninterruptible power supply (UPS), turn it off.
- Turn off the PC with the "POWER" button and remove the power cable from the outlet.
- Cover the keyboard with a cover to prevent dust from accumulating.

- Organize your workspace.
- Safety requirements in emergency situations:
- If you smell a burning or electric shock after turning on your computer, immediately unplug the computer and notify your supervisor.
- In the event of a fire, immediately start extinguishing it with the available fire extinguishing means and notify the city fire department at number 101 and the head of the fire protection department of the enterprise. Do not forget to extinguish electrical installations with carbon dioxide fire extinguishers, dry sand to avoid electric shock.
- In the event of an injury, stop work, provide first aid, call an ambulance at number 103, and if necessary, transport the victim to a medical facility.

Sequence of first aid:

1. Eliminate the effect of harmful factors that threaten the health and life of the victim (remove from the influence of electric current, evacuate from the contaminated atmosphere, extinguish burning clothes, etc.).
2. Assess the nature and degree of severity of bodily injuries, determine the greatest threat to the victim's life and take measures to save him.
3. As a matter of urgency, carry out the necessary rescue measures (restore the patency of the respiratory tract, perform artificial respiration, external heart massage, stop bleeding, immobilize fractures, apply bandages, etc.).
4. Support the basic vital functions of the victim until the arrival of medical personnel.
5. Call an ambulance or a doctor, or take measures to transport the victim to the nearest medical facility.

- Assistance to the victim by non-medical personnel should not replace medical assistance and should be provided only until the arrival of a doctor.
- Specific actions to provide first aid to a victim in case of various injuries are described in instruction No. 03-OP "On providing first (pre-medical) medical aid in accidents", which is studied by employees during the initial and subsequent safety briefings.

- In the event of other emergency situations, stop work and notify the supervisor.

This section deals with the measures that exclude or limit the influence of dangerous and harmful production factors on the technical staff of the IT department. The lighting calculation of the working area was carried out, as a result of which 9 lamps with 36 LED lamps were installed, which provide optimal lighting in the recommended norms of 300-500 lux. This lighting configuration provides a well-lit work area without violating established lighting standards. In addition, methodological guidelines for occupational health and safety when working with a personal computer and recommendations for fire safety in the IT department were presented.

CHAPTER 7
ENVIRONMENTAL PROTECTION

7.1. Analysis of the impact of man-made factors on the natural environment

As a result of intensive human activity, gradual changes took place in its environment, which led to the disruption of the biosphere and the formation of an artificial environment known as the technosphere. According to scientific data, today almost all areas where people live are man-made. The technosphere, created by man, covers almost the entire planet and even goes beyond it into space.

Technogenic environment, or technosphere, as a part of the environment, is the result of human activity caused by the influence of anthropogenic factors. In the technosphere, a person constantly performs at least two main functions: ensuring a comfortable stay in the living environment and creating and using protection systems against negative influences.

There are two forms of impact of the technosphere, namely direct and indirect impact on the environment and the human body [19].

7.2. The principle of operation of base stations and cellular devices and their negative impact on the environment

As a result of the rapid development of technologies and devices, avoiding the influence of electromagnetic radiation (EMF) in the modern world is practically an impossible task. Telephones and mobile base stations act as sources of EMF in this context. It is important to note that the impact of these sources on the human body is different.

One of the features of a mobile phone as a source of EMF is its maximum proximity to the user's head at a distance of two to five centimeters in uncontrolled conditions. This causes the effect of EMF on the brain, peripheral receptor zones of the vestibular and auditory analyzers, as well as on the retina of the eyes. The negative effects of mobile phone radiation also spread to surrounding people during a conversation.

Base stations generate electromagnetic fields impulsively, and their intensity depends on the time of day, coverage saturation, and the number of stations in the area. Since the base stations are located in the places of permanent human presence, there is a round-the-clock influence of a low-intensity electromagnetic field of the radio frequency range.

According to ecologists and hygienists, all ranges of electromagnetic radiation affect human health and performance with serious consequences. The electromagnetic field is considered more dangerous than radiation due to its widespread distribution. Electric fields of industrial frequency surround a person around the clock, due to radiation from electrical wiring, lighting devices, household electrical appliances and other sources.

According to research, the electromagnetic field can affect charged particles and currents, converting the energy of the field at the cell level into other types of energy. Cytogenetic studies have shown an increase in cells with chromosomal aberrations, and the electromagnetic field also affects tissues, causing changes in living tissues.

In turn, the effect of EMF on the nervous system can cause short-term memory impairment and other negative phenomena. The immune system also becomes a victim of the influence of electromagnetic radiation, which leads to disturbances in the processes of immunogenesis and other abnormalities.

In general, studies show that electromagnetic radiation affects various aspects of the human body, which requires attention and further scientific research.

The influence of the electromagnetic field on the reproductive system is expressed in a decrease in the function of spermatogenesis, a change in the menstrual cycle, a slowdown in embryonic development, the occurrence of birth defects in newborns and a decrease in lactation in nursing mothers.

The effect of the electromagnetic field on plants is well defined due to the results of numerous studies. Both weak and strong electromagnetic fields significantly affect the morphological, physiological, biochemical and biophysical characteristics of many plants, including their growth, development and reproduction.

Theoretically, electric field levels recorded near overhead lines are sufficient to damage plant leaves. Studies of the effect of electromagnetic radiation from power lines on plants confirmed a decrease in the dry weight of the above-ground mass of oats and sunflowers growing under the lines, compared to the control. The negative effect of the electromagnetic field on the potential nitrogenase activity of the soil rhizosphere population and the length of plant seedlings was also revealed.

Weak electromagnetic fields that do not reach the threshold of the thermal effect also affect changes in living tissues. Research on the biological impact of electromagnetic radiation from a cell phone, computer block and other electronic devices was conducted in various scientific centers. In the course of these studies, the harmfulness of electronic devices was evaluated both in the working and off states, including cases of lack of power [20].

The results of the conducted research on the influence of the cell phone, computer and other modern radio-electronic devices on various organisms, both in working and switched off conditions, turned out to be far from encouraging. The obtained data demonstrate the extremely negative impact of these technical means on biological objects, manifested in such aspects as:

- Increased mortality of microorganisms.
- Reduction of motor activity and survival of microorganisms.
- Reduction of energy potential in all important body systems.
- Violation of embryonic and larval development.
- Reduction of biochemical reactions and metabolic disorders.
- Deterioration of tissue regeneration processes.

7.3. Methods and means of protecting the environment from the influence of man-made factors

In order to reduce the impact of electromagnetic radiation (EMF) on personnel and the population in the area of influence of radio-electronic means, it is necessary to take a number of protective measures, which can be organizational, engineering-technical, and medical-prophylactic.

The responsibility for the implementation of organizational and engineering-technical measures rests primarily with the bodies of sanitary supervision. Together with the sanitary laboratories of enterprises and institutions that use sources of electromagnetic radiation, it is necessary to take measures for the hygienic assessment of new buildings and reconstruction of objects that produce and use radio equipment, as well as new technological processes and equipment using EMF, to conduct current sanitary supervision of objects that use radiation sources, and to carry out organizational and methodical work on training specialists and engineering and technical supervision.

At the design stage, such arrangement of irradiating and irradiated objects should be ensured, which would minimize the intensity of irradiation. Since it is impossible to completely avoid exposure, it is necessary to reduce the probability of people entering areas with high EMF intensity and reduce the time they are exposed to radiation.

Engineering and technical methods and means of protection, such as collective (a group of buildings, district, settlement), local (individual buildings, premises) and individual protection, are of great importance. Collective protection is based on the calculation of the propagation of radio waves in the conditions of a specific relief of the area and the effective use of natural screens, such as forest plantations or non-residential buildings.

Installation of antennas in high places allows to reduce the intensity of the electromagnetic field that irradiates the settlement. The orientation of the pattern, especially of highly directional antennas, can also reduce field strength. However, it is important to consider that a tall antenna can be more complex and expensive. The effectiveness of such protection also decreases with distance.

Local protection, based on the use of radio-protective materials, allows you to effectively protect the premises from radiation. Using highly conductive metal sheets and grids to reflect the field is an effective method. Pasting walls with metallized wallpaper, protecting windows with nets and metallized curtains are also effective protection measures [21].

Irradiation in such a room is reduced to a minimum, but the radiation reflected from the screens is dispersed in space and falls on other objects. For personnel who maintain radio equipment and are at a short distance, it is necessary to provide reliable protection by shielding the equipment. Along with reflective, widespread screens, materials that absorb radiation are used.

There is a large number of radio-absorbing materials - both homogeneous compounds and composite ones consisting of various dielectric and magnetic substances. In order to increase the efficiency of the absorbing surface, the screens are made rough, ribbed or with spikes. Radio-absorbing materials can be used to protect the environment from electromagnetic fields generated by a source located in a shielded object.

Personal protective equipment is used only in cases where other protective measures cannot be used or are not effective enough. This may be necessary when passing through zones of increased radiation intensity, during repair and adjustment work during accidents, as well as during short-term control and changes in radiation intensity. Such tools are inconvenient to use, limit the ability to perform work operations and worsen hygienic conditions.

Clothes made of metallized fabrics and radio-absorbing materials are used to protect the body. The metallized fabric consists of cotton or kapron threads wrapped with a metal wire, which weakens radiation by at least 20-30 decibels. When sewing parts of protective clothing, it is necessary to ensure contacts of insulated conductors. Electrosealing of seams is carried out with electrically conductive solutions or glues that provide galvanic contact or increase the capacitive connection of wires that are not in contact.

The eyes are protected by special glasses with glass, on the inner side of which there is a conductive tin dioxide film. The rubber frame of the glasses can have a pressed metal mesh or be covered with a metallized fabric, due to which microwave radiation is weakened by 20-30 decibels. The use of gloves and shoe covers is considered obsolete, since the permissible energy flow density for hands and feet is many times higher than for the body. Collective and individual means of protection can ensure long-term and safe work of personnel at radio protection facilities.

Measures for collective and individual protection against noise. A targeted fight against noise at the stage of its occurrence is considered the most effective method. For this, low-noise mechanical transmissions are being developed, methods of noise reduction in bearing units and fans are being studied.

Noise reduction through sound absorption. The noise-emitting object is placed in a casing lined with sound-absorbing material on the inner walls. It is important that the casing has sufficient sound-absorbing properties, does not interfere with the maintenance of the equipment and does not disturb the interior of the workshop. One of the variations of this method is a cabin in which the noisiest element is located and the operator also works. The inside of the booth is lined with sound-absorbing material to reduce the noise level inside, not just to isolate the source of the noise from the rest of the production space.

Noise reduction due to sound insulation. The essence of this method is to locate a noise-emitting object, or several of the noisiest objects, separately and isolated from the main, less noisy room with the help of a soundproof wall or partition. Noise protection is also achieved by installing noise screens and caps. This reduces the impact of noise on a smaller volume of space and the number of people.

Noise reduction due to the acoustic treatment of the room. Acoustic treatment involves covering the ceiling and the upper part of the walls with sound-absorbing material, which leads to a decrease in the intensity of reflected sound waves. Additionally, sound-absorbing shields, cones and resonator screens can be used. The effectiveness of such processing depends on the properties of the materials and structures used, their location, volume and geometry of the room, as well as the places where the noise sources are located. The method allows to reduce the noise level by 8 dBA.

It is important to consider noise reduction measures at the stage of designing industrial facilities and equipment. In particular, it is worth studying the possibility of placing noisy elements in separate rooms to reduce the number of employees who work in high noise. The work on noise reduction of existing equipment begins with the compilation of noise maps and analysis of noise spectra, based on which strategies for further actions are developed.

CONCLUSIONS

As a result of the scientific research practice, the use of Bluetooth and NFC in IoT was analyzed, the existing solutions, their approaches and errors were considered. As a result, a number of problems faced by users were revealed, namely the dependence of devices on the server. Existing solutions either do not cover all user needs, or are very local products that are not widely distributed.

The serverless approach provides seamless device connectivity, energy-efficient interaction, simplified implementation, and reduced infrastructure requirements. It provides centralized device management, over-the-air updates, and scalability to adapt to changing workloads. Additionally, the serverless architecture facilitates server-side integration, analytics, and analytics generation, unlocking the full potential of Bluetooth and NFC data for effective decision-making.

The target audience for IoT deployments using Bluetooth and NFC with a serverless approach include enterprises, consumers, healthcare providers, industrial sectors and urban planners. Every audience can benefit from the improved communication, data processing capabilities, and centralized management that this approach provides.

In summary, we can say that the combination of Bluetooth and NFC technologies with a serverless approach is a promising direction for the deployment of IoT, which allows solving existing problems and opening new opportunities in various industries. Further research and development in this area can contribute to the advancement of IoT applications, improved communication, automation and data analysis based on a smarter and more connected world.

LIST OF REFERENCE SOURCES

1. "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions" [Electronic resource] – URL https://www.mdpi.com/1424-8220/14/7/11706 (application date: 15.06.2023)
2. "Bluetooth Low Energy: A Technology Primer" [Electronic resource] – URL https://www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth_LE_Primer.pdf (application date: 15.06.2023)
3. "Near Field Communication (NFC) Technology and Measurements" [Electronic resource] – URL https://www.keysight.com/us/en/assets/3121-7387/application-notes/5992-1322.pdf (application date: 15.06.2023)
4. "Serverless Architectures" [Electronic resource] – URL https://www.manning.com/books/serverless-architectures (application date: 16.06.2023)
5. "Serverless Computing: A Comprehensive Overview" [Electronic resource] – URL https://www.researchgate.net/publication/331609725_Serverless_Computing_A_Comprehensive_Overview (application date: 17.06.2023)
6. "Number of users of smart homes worldwide from 2019 to 2028" [Electronic resource] – URL https://www.statista.com/forecasts/887613/number-of-smart-homes-in-the-smart-home-market-in-the-world (application date: 20.11.2023)
7. "Design and Implementation of a Framework for Smart Home Automation Based on Cellular IoT, MQTT, and Serverless Functions" [Electronic resource] – URL https://doi.org/10.3390/s23094459 (application date: 20.11.2023)
8. Arora S. Design of Secure IoT Systems: A Practical Approach Across Industries / S. Arora, R. Gambheer, M. Vohra. – New York: McGraw Hill, 2021. – 224 c.

9. Monk S. Raspberry Pi Cookbook: Software and Hardware Problems and Solutions / Simon Monk., 2022. – 650 с.

10. Heydon R. Bluetooth Low Energy: The Developer\'s Handbook / Robin Heydon., 2012. – 368 с.

11. Wang W. Pro iPhone Development with SwiftUI. Berkeley: Apress, 2022. URL: https://doi.org/10.1007/978-1-4842-7827-7 (application date: 31.05.2022).

12. Igoe T. Beginning NFC: Near Field Communication with Arduino, Android, and PhoneGap / T. Igoe, D. Coleman, B. Jepson., 2014. – 246 с.

13. Hagar J. D. IoT System Testing: An IoT Journey from Devices to Analytics and the Edge / Jon Duncan Hagar. – New York: Apress, 2022. – 348 с.

14. Guzman A. IoT Penetration Testing Cookbook: Identify vulnerabilities and secure your smart devices / A. Guzman, A. Gupta. – Birmingham: Packt Publishing, 2017. – 452 с.

15. DSanPiN 3.3.2-007-98 "Hygienic requirements for the organization of work with visual display terminals of electronic computing machines"

16. ГН 3.3.5-8-6.6.1-2002 "Hygienic classification of work according to indicators of harmfulness and dangerous factors of the production environment, difficulty and tension of the labor process"

17. NPAOP 0.00-1.29-97 "Rules for protection against static electricity"

18. NAPB A.01.001-2004 "Rules of fire safety in Ukraine"

19. Forecasting environmental risks using hierarchy analysis and fuzzy set theory: international scientific and practical conference "I All-Ukrainian Congress of Ecologists": Abstracts of reports. Ukraine, Vinnytsia, October 4-7 2016. – 2016. – С.25.

20. Klap Y. A., Yaremkevich O. S., Chervetsova V. G., Zayarniuk N. L., Novikov V. P., Research on the impact of electromagnetic, permanent magnetic and acoustic fields on the human body // Visnyk Nats. Lviv Polytechnic University. – 2016 – № 812. – С. 365–372.

21. The current state of research on the impact of electromagnetic radiation on the human body [Electronic resource]/[A. P. Chornyi, V. V. Nikiforov, D. I. Rodkin, V. Yu.

Nozhenko] // Engineering and educational technologies in electrical and computer systems: a quarterly scientific and practical journal. – Kremenchuk: KrNU, 2013.