Ministry of Education and Science of Ukraine
Ternopil Ivan Puluj National Technical University

Faculty Of Computer Information Systems and Software Engineering
(Full name of faculty)
Computer science Department
(Full name of department)

QUALIFYING PAPER

For the degree of

master
(Degree name)

topic: Design and modeling of passwordless authentication architectures in compliance with industry-standard security frameworks

Submitted by: sixth year student, group ICTm-62
Specialty: 126 Information Systems and Technologies
(Code and name of specialty)

Akinyemi Victor Oluwaseyi
(signature) (Surname and initials)

Supervisor Holotenko O.S.
(signature) (Surname and initials)

Standards verified by Nykytiuk V.V.
(signature) (Surname and initials)

Head of Department Bodnarchuk I.O.
(signature) (Surname and initials)

Reviewer Tysh I.V.
(signature) (Surname and initials)

Ternopil
2026

Ministry of Education and Science of Ukraine
Ternopil Ivan Puluj National Technical University

Faculty Of Computer Information Systems and Software Engineering
(Full name of faculty)
Department Computer science Department
(Full name of department)

APPROVED BY
Head of Department
Bodnarchuk I.O.
(signature) (Surname and initials)
« » 20__ p.

ASSIGNMENT

for QUALIFYING PAPER

for the degree of *master*
(Degree name)
specialty 126 Information Systems and Technologies
(Code and name of the specialty)
student Akinyemi Victor Oluwaseyi

1. Paper topic: Design and modeling of passwordless authentication architectures in compliance with industry-standard security frameworks

Paper supervisor: Holotenko Oleksandr Serhiyovych,, PhD, Assoc. Prof.
(Surname, name, patronymic, scientific degree, academic rank)

Approved by university order as of «___» _____ 20__ №____.
2. Student's paper submission deadline: _____
3. Initial data for the paper Industry-standard specifications for passwordless authentication (WebAuthn, FIDO2, CTAP), cryptographic methods based on public key infrastructure, hardware and platform authenticators
4. Paper contents (list of issues to be developed) *1. P*roblem statement for modeling passwordless authentication systems based on industry standards. *2. M*ethods and tools for developing passwordless authentication systems. *3. Implementations of a passwordless authentication system. 4. Developments of the startup project. 5. Occupational safety and health.*
5. List of graphic material (with exact number of required drawings, slides)

6. Advisors of paper chapters.

Chapter	Advisor's surname, initials and position	Signature, date	
		assignment was given by	assignment was received by
Occupational Safety			
Safety in Emergency Situations			

7. Date of receiving the assignment.

TIME SCHEDULE

LN	Paper stages	Paper stages deadlines	Notes
1	Literature review on the topic of the master's thesis		
2	Analysis of existing solutions in the field of passwordless		
3	Comparative analysis of passwordless authentication methods		
4	Design of the architectural model of a passwordless		
5	Software development based on the proposed architectural		
6	Testing of the developed software solution		
7	Preparation of master's thesis materials		
8	Pre-defense of the master's thesis		
9	Defense of the master's thesis		

Student _____ Akinyemi Victor Oluwaseyi
(signature) (surname and initials)

Paper supervisor _____ Holotenko O.S.
(signature) (surname and initials)

ANNOTATION

Design and modeling of passwordless authentication architectures in compliance with industry-standard security frameworks // The educational level "Master" qualification work // Akinyemi Victor Oluwaseyi // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, ISTm-62 group // Ternopil, 2026 // P. 95, fig. – 25, tables – 28, annexes – 1, ref. – 40.

First chapter analyzes the limitations of password-based authentication and reviews industry standards such as FIDO2 and WebAuthn as a foundation for passwordless systems.

Second chapter describes cryptographic principles, authentication algorithms, and software tools used to design and implement passwordless authentication solutions.

Third chapter presents the architecture, server-side implementation, database design, user interfaces, and testing of the WebAuthn-based authentication system.

Fourth chapter examines the startup concept, technological feasibility, market opportunities, and commercialization strategy of the proposed passwordless authentication solution.

Fifth chapter analyzes workplace conditions and potential occupational risks to ensure safe and compliant working environments.

Object of the research – the process of user authentication in modern information systems using passwordless security mechanisms.

Subject of the research – architectural models, methods, and technologies for implementing passwordless authentication based on industry standards such as WebAuthn and FIDO2.

Keywords: passwordless authentication, information security, biometric data, cryptography, security tokens, user data, cybersecurity.

CONTENTS

LIST OF ABBREVIATIONS, SYMBOLS, AND TERMS

FIDO (Fast Identity Online) is a set of technical specifications for user authentication, commonly used in scenarios such as fingerprint-based login or multi-factor authentication.

WebAuthn (Web Authentication) is a web standard developed by the World Wide Web Consortium (W3C) in collaboration with the FIDO Alliance, enabling websites and online services to authenticate users using public key cryptography.

CTAP (Client to Authenticator Protocol) is a specification developed by the FIDO Alliance that defines how external authenticators, such as hardware security keys or mobile devices, communicate with client platforms, including web browsers and operating systems.

TPM (Trusted Platform Module) is a dedicated microcontroller designed to secure hardware by storing and managing cryptographic keys within a trusted execution environment.

YubiKey is a hardware authentication device developed by Yubico to enhance the security of computers, networks, and online services.

MitM (Man-in-the-Middle) is a type of cyberattack in which an attacker intercepts and potentially alters communication between two parties who believe they are communicating directly with each other.

Authenticator is a device or software application that verifies a user’s identity without requiring a traditional password.

INTRODUCTION

Modern methods for protecting information systems are a fundamental prerequisite for the secure functioning of the digital society, particularly in the context of the growing number of cyber threats. The protection of personal data and confidential information has become a priority task, stimulating the search for new approaches in the field of authentication. One of the most promising directions is the use of passwordless authentication systems, which significantly enhance the security of user accounts. A key instrument in this process is the WebAuthn standard, designed to provide secure authentication through the use of asymmetric cryptography and biometric technologies. This standard offers an effective alternative to traditional passwords, which have numerous drawbacks, including vulnerability to phishing attacks and the risk of unauthorized use.

Traditional password-based authentication methods are frequently exposed to attacks such as interception or brute-force guessing, enabling unauthorized access to user accounts. In this regard, the implementation of passwordless authentication technologies represents a critical task in the field of cybersecurity. Although the FIDO2 and WebAuthn standards have significantly expanded the capabilities of passwordless authentication, their effective integration into real-world information systems requires further research, particularly with respect to compatibility with existing infrastructures, where technical challenges may arise.

The purpose of the master's thesis entitled "Design and modeling of passwordless authentication architectures in compliance with industry-standard security frameworks" is to develop software solutions for passwordless authentication in information systems in order to enhance the security of user data.

To achieve this goal, the following **research objectives** were defined:

- to study existing passwordless authentication standards;

- to analyze architectural solutions for integrating WebAuthn into modern information systems;

- to design a passwordless authentication model;

- to implement a prototype system based on the proposed model.

The objective of the thesis is a development of software tools for

passwordless authentication in information systems to improve the security of user data.

Research tasks are to investigate existing standards of passwordless authentication, analyse architectural solutions for implementing WebAuthn in modern information systems, develop passwordless authentication models and implement a test system with the model implementation.

Research methods. The research employed experimental modeling to test the passwordless authentication system with various tokens in near-real conditions. Architectural modeling was applied to assess the compatibility of the WebAuthn standard with existing systems. Empirical testing with real tokens was conducted to verify authentication reliability. Security analysis identified the strengths and risks of WebAuthn and FIDO2 standards, while performance analysis evaluated the system’s efficiency in processing requests.

The scientific novelty of the obtained results lies in the development and implementation of an architectural model for a passwordless authentication system based on the WebAuthn standard, ensuring enhanced user data security and reducing the risks associated with traditional passwords.

The practical significance of the obtained work results lies in implementing a passwordless authentication system based on the WebAuthn standard, which increases user data security and reduces password-related risks. The developed architectural model and implemented server side can be integrated into modern information systems, ensuring compatibility with existing infrastructures and enhancing overall.

Approbation of the results. The main provisions of this work were discussed at: III International Scientific and Practical Conference "Technology development: shaping modern thinking and scientific approaches", Krakow, Poland, January 19–21, 2026.

Publications. The main results of the qualification work were published at the conference (See Appendix A).

Structure and scope of qualification work. The total length of the dissertation is 95 pages, including 89 pages of main text, 25 tables, 28 figures, and 4 pages of references totaling 40 sources.

1. PROBLEM STATEMENT FOR MODELING PASSWORDLESS AUTHENTICATION SYSTEMS BASED ON INDUSTRY STANDARDS

This chapter examines the key issues associated with the use of passwords in modern information systems and substantiates the necessity of transitioning to passwordless authentication methods. In addition, it provides an overview of existing passwordless authentication technologies based on industry standards such as WebAuthn and FIDO2. Furthermore, ready-made solutions built upon these standards are analyzed. The purpose of this chapter is to formulate the core requirements for passwordless authentication systems and to define the tasks for modeling such systems in accordance with contemporary security standards.

1.1 Password-related issues in modern information systems

In today's digital environment, passwords remain one of the most widely used authentication mechanisms; however, their use introduces a number of significant challenges that create security risks for both users and organizations. The primary issues associated with passwords include their vulnerability to various types of attacks, poor usability, and the necessity to manage a large number of user accounts.

First, vulnerability to attacks represents one of the most serious threats to password security. Under modern conditions, attackers employ a wide range of techniques to obtain passwords, including phishing, brute-force attacks, and data interception [1]. Phishing attacks, in particular, are highly prevalent and enable adversaries to acquire confidential information by deceiving users through fraudulent websites or email messages. Such attacks frequently result in data breaches, which may have severe consequences for individuals as well as large organizations.

Another critical issue is password reuse across multiple information systems. According to Google research, more than half of users reuse identical or similar passwords for multiple accounts, which significantly increase the risk of compromise [2]. Once a single account is breached, attackers can easily gain access to other services protected by the same credentials. This practice is especially dangerous in

corporate environments, where access to sensitive resources may be obtained through a compromised personal account.

In addition to security concerns, passwords pose substantial usability challenges and negatively affect the user experience. Users are often required to memorize a large number of complex passwords for different services, which leads to the creation of weak or predictable passwords. This also encourages insecure practices such as writing passwords down or storing them in unprotected locations.

Regular password changes, which are recommended by many organizations, further reduce usability. Frequent password updates often result in users making minor variations of previous passwords or recording new passwords to avoid forgetting them, ultimately decreasing overall system security.

From an organizational perspective, password management constitutes an additional challenge. To ensure an adequate level of security, organizations must enforce password complexity policies, frequent updates, and multi-factor authentication. Nevertheless, even under these measures, account compromise due to stolen or leaked passwords remains a widespread threat.

Finally, the economic aspect represents an important factor. The costs associated with recovering from data breaches caused by lost or stolen passwords can be substantial. Cybersecurity incident reports consistently identify passwords as a weak link leading to large-scale data breaches and reputational damage.

Thus, the problem of passwords in modern information systems is significant from both cybersecurity and usability perspectives. Passwordless authentication solutions such as WebAuthn and FIDO2 introduce new opportunities to protect systems against malicious actors by increasing security and reducing reliance on the human factor. The adoption of these technologies represents a step toward building more secure and user-friendly authentication systems, making this topic highly relevant for research and practical implementation.

1.2 FIDO Standard

The FIDO (Fast Identity Online) standard is a key technology for passwordless authentication that aims to reduce reliance on passwords and enhance security in

information systems. Established in 2012, the FIDO Alliance brought together major industry stakeholders to develop open and interoperable technical specifications enabling the use of biometric authentication methods and public key cryptography. The standard is based on the premise that passwords are inherently insecure and inconvenient and should therefore be replaced with more secure and user-friendly alternatives. The operating principles of FIDO authentication keys are illustrated in Figure 1.1.

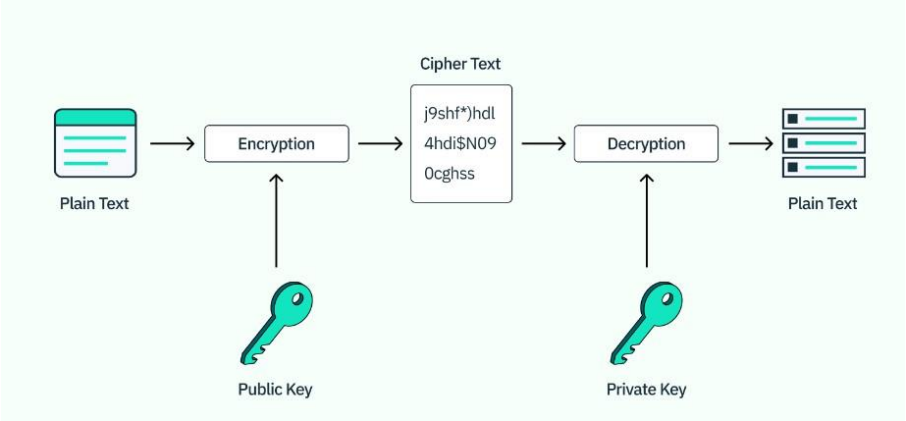

Figure 1.1 – Abstract diagram of authentication key operation in the FIDO standard

As illustrated in Figure 1.1, authentication based on the FIDO standard relies on a public-private key pair, where the private key is securely stored on the user's device, while the public key is transmitted to the server. The private key is used to sign cryptographic challenges generated by the server in order to verify the user's authenticity. Through this mechanism, the server can confirm that the user is who they claim to be without the need to store passwords on the server side.

Since the private key is stored locally on the user's device or within a hardware security module, the authentication process becomes localized and significantly more resistant to remote attacks. To confirm their identity, users may employ biometric data such as fingerprints or facial recognition, which enhances both usability and security.

The FIDO Alliance not only develops technical specifications but also certifies products that implement these standards, ensuring their quality and compliance with security requirements. Furthermore, all FIDO specifications are submitted for standardization through international organizations, which contributes to the

widespread adoption of passwordless authentication among developers and enterprises.

1.3 WebAuthn Standard

WebAuthn (Web Authentication) is a standard developed by the World Wide Web Consortium (W3C) that enables secure user authentication without the use of passwords.

WebAuthn allows web applications to authenticate users through asymmetric cryptographic keys instead of traditional passwords. The authentication process is based on the generation of a key pair consisting of a private and a public key. The private key is stored on the user's device, such as a hardware token or security key, while the public key is transmitted to the server.

The user registration process involves the creation of new credentials based on the provided username. This mechanism forms the foundation of passwordless login supported by WebAuthn and is illustrated in more detail in Figure 1.2.

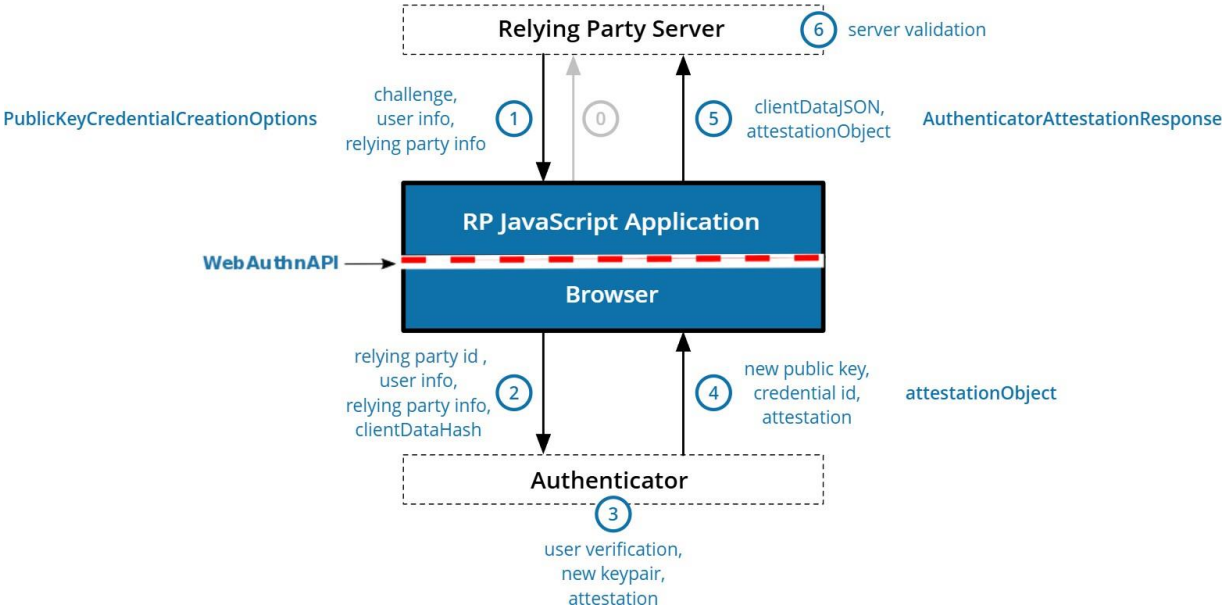

Figure 1.2 – Sequence diagram of the WebAuthn-based registration process

Let us examine the sequence of steps illustrated in the diagram:

1. Initiation of the registration process. The user visits a WebAuthn-enabled website and clicks the "Register" button. This action initiates a request to the

authentication server (Relying Party), which is responsible for handling authenticator registration requests.

2. Server-side challenge generation. The server generates a random data set known as a “challenge,” which ensures the uniqueness of the request. Along with the challenge, the server constructs a PublicKeyCredentialCreationOptions object containing information about the server (e.g., domain name), the user identifier, supported cryptographic algorithms, timeout parameters, and authenticator preferences. This object is sent to the user’s browser.

3. Data transfer to the browser. The browser receives the data from the server and forwards it to a local authenticator or an external hardware device (e.g., YubiKey, Touch ID, Face ID).

4. Interaction with the authenticator. The authenticator requests user confirmation. For example, the user may enter a PIN, provide a fingerprint scan, or use facial recognition. This step ensures that the operation is explicitly authorized by the user.

5. Key pair generation by the authenticator. After successful verification, the authenticator generates a cryptographic key pair. The private key remains securely stored on the user’s device, while the public key and additional attestation data are returned to the browser for further processing.

6. Attestation objects formation. The authenticator constructs an attestation object (attestationObject) containing the public key, information about the authenticator (type, model, manufacturer), and a digital signature. This object, together with browser data (clientDataJSON), is transmitted to the server.

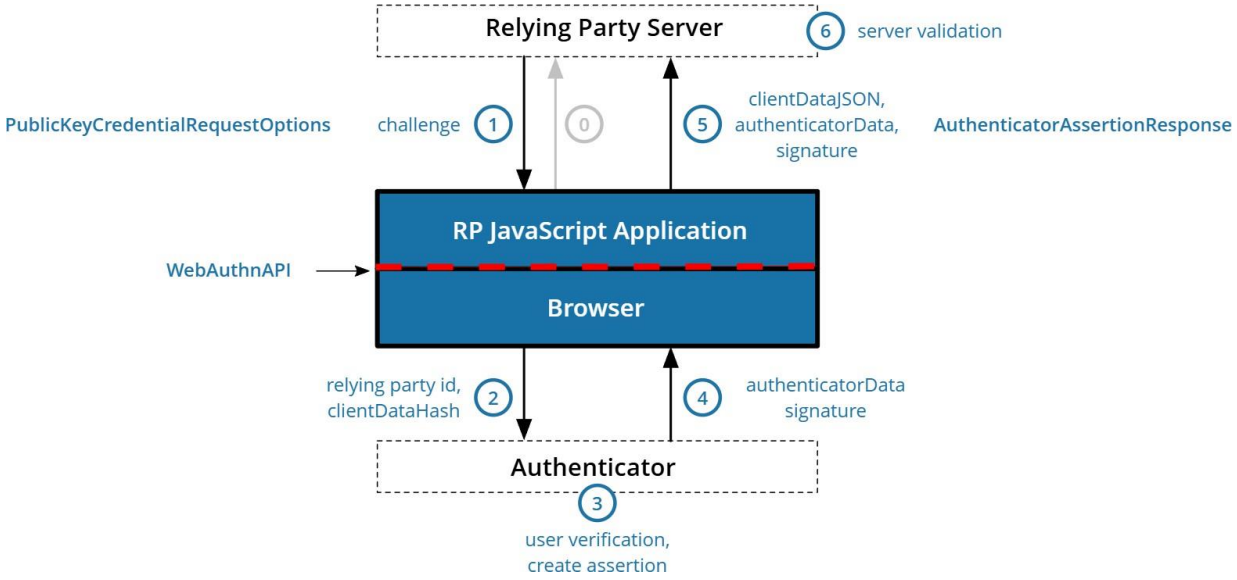

Figure 1.3 – Abstract representation of the authentication process based on the WebAuthn standard

Let us examine the sequence of steps illustrated in the diagram:

1. Initiation of the authentication process. The user visits a website and clicks the “Log In” button. This action generates a request to the authentication server (Relying Party), which verifies whether the user account exists and initiates the authentication procedure.

2. Server-side challenge generation. The server generates a random data set known as a “challenge,” which ensures request uniqueness and prevents replay attacks. The server also sends user account information (e.g., the identifier of the registered authenticator) together with a PublicKeyCredentialRequestOptions object, which includes: challenge – a unique value for the current session; rpId – the server identifier (domain); allowCredentials – a list of authenticator identifiers that may be used for authentication; timeout – the response time limit; userVerification – the required level of user verification (e.g., biometrics or PIN).

3. Data transfer to the client. The server transmits the PublicKeyCredentialRequestOptions object to the browser or client application. The browser invokes the authenticator via the WebAuthn API.

4. Request to the authenticator. The browser forwards the received data to a local authenticator or an external device (e.g., YubiKey, Touch ID, and Face ID). The authenticator identifies the user and requests explicit confirmation.

5. User verification by the authenticator. The user confirms their identity using biometric data (fingerprint, facial recognition, etc.), a PIN code, or a physical action (e.g., touching a YubiKey).

6. Signature generation by the authenticator. The authenticator uses the registered private key to generate a digital signature based on the server challenge, client data (clientDataJSON), and other session-specific parameters.

7. Returning the response to the client. The authenticator sends the signed data to the browser together with additional information: authenticatorData, signature, and clientDataJSON.

8. Transmission of the response to the server. The browser sends this data back to the server. The server receives a PublicKeyCredential object containing the authenticator identifier (id), signature data (authenticatorData, signature), and client JSON data (clientDataJSON).

9. Server-side signature verification. The server compares the received challenge with the value generated at session initiation, verifies the digital signature using the public key stored during registration, and analyzes authenticatorData to confirm response validity.

10. Completion of authentication. If all checks are successful, the server confirms user authentication and grants access to the system or protected resources.

After analyzing the diagrams that illustrate the structure and processes of WebAuthn-based authentication, it is important to consider the general advantages and limitations of this standard.

WebAuthn offers significant benefits, particularly in terms of security and usability. The most notable advantage is the use of asymmetric cryptography, which provides a high level of protection. Since the private key is stored on a hardware authenticator and the public key is transmitted to the server, account access remains protected even in the event of server compromise [4]. In addition, WebAuthn significantly increases resistance to phishing attacks, as the domain required for authentication is bound to the authenticator rather than a virtual server, preventing domain spoofing [5]. The standard also reduces the risks associated with confidential data leakage, as only public keys are stored on the server and biometric information is never transmitted or stored remotely [6].

An additional advantage is compatibility with a wide range of browsers and devices, ensuring flexibility and avoiding vendor lock-in. Users may choose their preferred authentication methods, such as fingerprints or hardware tokens, which enhances convenience and user control. Moreover, the absence of passwords reduces cognitive burden, as users no longer need to remember or manage complex credentials.

Despite its much strength, WebAuthn also exhibits certain limitations. Limited cross-device portability represents one of the primary challenges, users cannot easily migrate credentials between authenticators. Consequently, the loss or damage of an authenticator may result in serious account recovery issues. Furthermore, the cost of deployment may be prohibitive for some organizations, as it requires dedicated hardware and qualified technical personnel. The standard also demands a high level of technical competence from both users and organizations to ensure correct and secure operation.

This chapter examined the key issues associated with passwords in modern information systems, including their vulnerabilities and the risks arising from reliance on this authentication method. Passwords have demonstrated unreliability due to susceptibility to phishing, brute-force attacks, and reuse, posing significant threats to data security for both individuals and organizations. These challenges underscore the necessity of adopting alternative authentication methods that provide both usability and enhanced protection.

The analysis of modern standards such as FIDO2 and WebAuthn demonstrates their potential to address these issues. By employing asymmetric cryptography, these technologies enable private keys to be stored on user devices, eliminating dependence on passwords and reducing the risk of data compromise. Particular attention was given to WebAuthn, which ensures compatibility with a wide range of browsers and devices, allowing users to authenticate using biometrics or hardware tokens.

Although the adoption of passwordless methods requires addressing challenges related to cross-device compatibility, implementation costs, and technical complexity, the benefits, resilience to phishing, improved usability, and data confidentiality, clearly outweigh these obstacles.

Thus, passwordless authentication based on industry standards represents a

critical step toward the development of secure and user-friendly information systems. Further research and modeling of such systems will contribute to their effective integration into modern platforms and promote their widespread adoption.

2 METHODS AND TOOLS FOR DEVELOPING PASSWORDLESS AUTHENTICATION SYSTEMS

This chapter examines the primary methods and tools used for developing passwordless authentication systems. Since security and user convenience are critical aspects of modern information systems, it is essential to employ advanced technologies that eliminate passwords as a weak link. Passwordless authentication is based on cryptographic techniques, such as the use of public and private keys, as well as modern authentication mechanisms, including biometric data and hardware tokens [7]. The development of such systems requires both mathematical methods to ensure security and software tools for implementing these methods.

2.1 Fundamentals of Passwordless Authentication

The foundation of any passwordless authentication system is asymmetric cryptography, which enables reliable authentication without the need for passwords. Passkeys represent an approach based on a cryptographic key pair: a private key stored on the user's device and a public key stored on the server. During authentication, the private key is used to sign a request, and the server verifies the signature using the corresponding public key. This mechanism ensures that user data remain protected throughout all stages of interaction.

Passkeys can be divided into two main types:

1. Synced passkeys. These are stored on the user's device and synchronized across multiple devices via secure cloud services. This allows users to employ the same passkey on different devices while maintaining both convenience and security [8].

2. Device-bound passkeys. These are stored locally on a specific device and are not synchronized with others. They provide a higher level of security but restrict usage to a single device [9].

In the context of passwordless authentication, two categories of authenticators are used:

1. Platform authenticators. These are built into the user's device, such as

fingerprint scanners or facial recognition systems on smartphones and laptops [10]. They provide convenient authentication without additional hardware but are limited to a specific platform.

2. Roaming authenticators. These are external devices, such as hardware security tokens (e.g., YubiKey), which can be connected to different devices via USB, NFC, or Bluetooth. They offer a high level of security and universality, enabling authentication across multiple devices and platforms [11].

The private key is confidential and never leaves the user's device, significantly reducing the risk of compromise. In WebAuthn- or FIDO2-based systems, the key is stored either on a hardware token (e.g., YubiKey) or within a secure enclave of the device responsible for protecting cryptographic material and other sensitive data, such as the Trusted Platform Module (TPM) on computers or Secure Enclave on smartphones [12].

The public key is transmitted to the server during user registration. When authentication is initiated, the server generates a random data set and sends it to the user's device. The device signs this data using the private key and returns the signature to the server. The server verifies the signature using the stored public key and, if valid, grants access to the user.

This approach provides strong protection, as even in the event of server compromise, attackers cannot obtain users' private keys. Consequently, passwordless authentication systems are resilient against many types of attacks, including phishing and man-in-the-middle attacks [13].

Furthermore, biometric data such as fingerprints or facial recognition may be used for user verification. Importantly, these data are stored exclusively on the user's device and are never transmitted to the server, providing an additional layer of protection [14].

The advantages of asymmetric cryptography in passwordless authentication include:

- protection against phishing, since attackers cannot forge a valid signature even if the public key is known;

- protection against credential theft, as no passwords are used in the authentication process;

- enhanced security, because private keys remain on user devices even if the server is compromised.

This architecture makes asymmetric cryptography an ideal foundation for passwordless authentication, enabling increased security and improved user experience in modern information systems.

Passwordless authentication systems may also employ the “magic link” method. Under this approach, after a user enters an email address or username on the login page, the system sends a one-time link to the specified address [15]. By following this link, the user is automatically authenticated without entering a password. This method simplifies authentication and can be used either as a primary login mechanism or as an additional authentication factor [16]. However, its security depends on the reliability of the user’s email account, as compromise of the mailbox may lead to unauthorized access. A comparison between passkey-based authentication and magic links is presented in Table 2.1.

Table 2.1 - Comparison of Passwordless Authentication Methods

Parameter	Magic Links	Passkeys
Authentication mechanism	A one-time link sent via email or SMS	Asymmetric cryptography using a private-public key pair
User convenience	High; does not require remembering passwords, but depends on access to email or phone	High, especially when using biometrics; does not require access to additional devices
Security	Moderate; depends on the security of email or phone and may be vulnerable to man-in-the-middle attacks	High; resistant to phishing and man-in-the-middle attacks due to cryptographic protection

End of Table 2.1

Use of biometrics	Not supported	Supported; includes fingerprints, facial recognition, etc.
Compatibility	Broad; does not require specialized hardware	Requires support from devices and services; gradually being adopted
Validity period	Limited; links typically have a short lifespan	Persistent; keys are stored on the device and can be reused multiple times

Although magic links provide a convenient passwordless authentication mechanism, they exhibit certain limitations in terms of security and their dependence on access to email or mobile devices. Passkeys, by contrast, leverage asymmetric cryptography and support integration with biometric methods, thereby offering a higher level of security and usability. They are inherently resistant to phishing attacks and do not require access to additional devices or external services during authentication. Furthermore, passkeys can be synchronized across multiple user devices, enhancing overall convenience. Consequently, for the development of modern passwordless authentication systems, passkeys represent a more reliable and forward-looking solution.

2.2 Algorithms and Technologies for Implementing Passwordless Authentication

One of the most widely adopted standards for implementing passwordless authentication is FIDO2, which encompasses the WebAuthn and CTAP (Client to Authenticator Protocol) technologies. WebAuthn enables authentication within web

applications and browsers through asymmetric cryptography, whereas CTAP facilitates the use of hardware tokens and biometric mechanisms for user verification.

FIDO2 supports both hardware security keys (e.g., YubiKey) and biometric methods such as fingerprint scanning and facial recognition, significantly enhancing user convenience while maintaining a high level of security. It is important to note that authentication using these methods is inherently protected against phishing attacks, since the domain associated with the login process is bound to the authenticator or device rather than stored on a remote server.

In addition, FIDO2 provides scalability and cross-platform compatibility, allowing seamless integration into existing systems and applications. This makes FIDO2 an attractive solution for organizations of any size seeking to improve both the security and usability of user authentication.

2.3 Software Tools for Developing Passwordless Authentication Systems

A wide range of software tools is employed in the development of such systems, enabling the integration of passwordless authentication into modern information systems.

2.3.1 Java Programming Language

The Java programming language was selected for the development of this project due to its numerous advantages, which make it an ideal choice for building reliable and scalable applications. Java is a high-level, object-oriented programming language developed by Sun Microsystems in 1995 [17]. One of its key characteristics is platform independence, which allows applications to run on different operating systems without modification. This is achieved through the Java Virtual Machine (JVM), which interprets byte code independently of the underlying platform.

Java was chosen for development based on several key advantages:

- platform independence. Owing to the JVM, Java code can be executed on any operating system without modification;

- object-oriented paradigm. Java supports object-oriented programming,

promoting modularity, code reuse, and maintainability;

- security. Java incorporates built-in security mechanisms that protect against many common vulnerabilities, such as buffer overflows;

- extensive ecosystem. Java provides a rich set of libraries, frameworks, and tools that accelerate the development process.

Compared to other programming languages such as C++ and Python, Java exhibits several distinctive advantages. Unlike C++, Java provides automatic memory management through garbage collection, reducing the risk of memory leaks and improving application stability [18]. While Python is renowned for its simplicity, Java offers a stricter type system, which facilitates early error detection during compilation and contributes to greater code reliability.

To better illustrate the position of Java among other programming languages, a comparative overview is presented in Table 2.2.

Table 2.2 – Comparison of Java with Other Programming Languages

Characteristic	Java	C++	Python
Paradigm	Object-oriented	Object-oriented, procedural	Object-oriented, procedural
Platform independence	High (via JVM)	Low (compiler-dependent)	High (interpreted language)
Memory management	Automatic (garbage collection)	Manual	Automatic (garbage collection)
Execution speed	High	Very high	Moderate
Syntax complexity	Moderate	High	Low

End of Table 2.2

Primary application domains	Web development, mobile applications, enterprise systems	System programming, games, drivers	Web development, scientific computing, machine learning

Java is also characterized by high performance due to compilation into byte code and the use of Just-In-Time (JIT) compilation, which enables runtime optimization of program execution [19]. Although Java may be slower than C++ in certain computationally intensive tasks, its performance is sufficient for the majority of enterprise applications. In this work, Java was used as the primary programming language, as it can be seamlessly integrated with frameworks such as Spring and Hibernate.

2.3.2 Spring Boot Framework

Spring Boot is a micro services-oriented framework based on the Spring Framework that significantly simplifies its usage through convention-over-configuration and automatic configuration principles. Spring Boot was designed to accelerate development and reduce the complexity of building modern Java applications. Its primary objective is to provide a ready-to-run foundation without the need for extensive manual configuration [20]. This has made it extremely popular among developers who aim to rapidly build applications for the web, cloud platforms, and micro services [21].

One of the key advantages of Spring Boot is its ability to integrate with other widely used technologies. The framework supports most relational databases, such as PostgreSQL, MySQL, and H2, as well as non-relational databases including MongoDB and Redis. By leveraging Spring Data, developers can focus on business logic instead of writing complex SQL queries. In addition, its support for RESTful APIs, integration with Hibernate, and modules for cloud computing make it highly

flexible [22].

Another important feature of Spring Boot is its integration with Spring Initializr. This online tool allows developers to generate a base project within minutes by selecting the required dependencies and structure. Spring Boot also includes an embedded web server (Tomcat or Jetty), enabling applications to be launched directly from a standard JAR file. This significantly simplifies deployment and testing in both local and cloud environments.

Spring Security serves as the core security module within Spring Boot, providing mechanisms for authentication and authorization. It supports integration with FIDO2 and WebAuthn, enabling passwordless authentication through cryptographic keys and hardware tokens. Using Spring Security, developers can implement multi-factor authentication based on tokens such as YubiKey or biometric data, thereby enhancing application security. The framework allows the configuration of diverse security policies, which is essential for implementing complex authentication systems [23].

In this thesis, Spring Boot was employed to streamline and accelerate development by providing automatic configuration and an embedded server, enabling rapid application creation and deployment. Spring Security was integrated to ensure data protection and access control by implementing authentication and authorization mechanisms.

2.3.3 Yubico Library

Yubico provides libraries and SDKs for working with hardware tokens such as YubiKey, which are among the key instruments for implementing passwordless authentication. YubiKey is a hardware device that stores private cryptographic keys and enables user authentication based on public-key cryptography. The Yubico library allows seamless integration of these tokens into Java applications, simplifying development and configuration of cryptographic operations. In addition, Yubico libraries support multi-factor authentication (2FA), where a physical device is used to confirm login.

Yubico libraries are widely used in enterprise solutions for data protection.

They enable strong authentication for access to internal systems, email services, or VPNs. This is particularly valuable for organizations handling sensitive information and seeking to minimize the risk of data breaches [24].

In the startup project, the Java-based Yubico library enabled implementation of the core passwordless authentication logic, including cryptographic key management.

2.3.4 Hibernate Framework

Hibernate is a widely used object-relational mapping (ORM) framework for Java that allows developers to interact with databases using an object-oriented approach [25]. Its primary objective is to simplify the development of applications that operate on large volumes of data [26].

Key features of Hibernate include:

- automatic ORM, enabling Java objects to be mapped directly to database tables through annotations or XML configuration;

- Hibernate Query Language (HQL), which resembles SQL but operates on objects rather than tables, making queries more intuitive for Java developers;

- caching mechanisms, including first-level (session-local) and second-level (global) caches, which reduce database queries and improve performance;

- support for numerous databases, such as PostgreSQL, MySQL, Oracle, and SQL Server, with seamless integration into Spring Data;

- object lifecycle management, providing states such as Transient, Persistent, and Detached, allowing flexible data control and consistency.

Hibernate offers several advantages for developers. It significantly reduces the amount of manual SQL code, provides portability across different databases through HQL, manages transactions automatically, and integrates easily with frameworks such as Spring. These characteristics make it particularly suitable for complex enterprise systems.

In this project, Hibernate was used for object-relational mapping, enabling seamless interaction between Java objects and database tables. This simplified transaction management and query execution while maintaining an object-oriented approach to data handling.

2.3.5 Maven Build Tool

Maven is a widely used tool for dependency management and builds automation in Java projects. It facilitates integration of libraries such as Spring Security, Yubico SDK, and Hibernate, allowing developers to easily manage configurations and updates. Maven simplifies the setup and maintenance of passwordless authentication systems by automatically handling dependencies and project configuration [27].

Maven is based on the Project Object Model (POM), an XML file containing all project information and configuration details. The POM defines the project structure, versioning, and dependencies. When a task is executed, Maven locates the POM in the current directory and resolves the required components. Maven automatically downloads the appropriate JAR files, ensuring correct versions are used without manual intervention [28].

Maven offers numerous advantages, including automation of building, documentation, packaging, and distribution processes. Adding new dependencies requires only a simple entry in the pom.xml file. However, Maven also has limitations, such as the need for IDE plugins and occasional performance overhead.

In this work, Maven served as the dependency management and build automation system. It simplified integration of external libraries and frameworks, provided a standardized project structure, and enabled convenient version control of dependencies.

2.3.6 PostgreSQL Database

PostgreSQL is a powerful, scalable open-source database management system that combines rich functionality, reliability, and a high level of security. By adhering to the ACID principles (Atomicity, Consistency, Isolation, Durability), PostgreSQL guarantees data integrity and is among the best choices for projects requiring stability and accuracy. Its support for complex data types such as JSON, XML, and geospatial data makes it especially suitable for modern applications [29].

In the context of passwordless authentication systems, PostgreSQL is used to store users' public cryptographic keys, associated metadata, and other information required for authentication. This enables scalable user management and efficient verification during login. Advanced indexing mechanisms such as GiST and GIN provide fast data access even with large datasets [30].

With built-in encryption mechanisms such as TLS/SSL, PostgreSQL ensures data protection during transmission. It also supports advanced access control mechanisms, including role-based models and granular privileges at table, row, and column levels, adding an additional layer of security against unauthorized access.

PostgreSQL's flexibility allows seamless integration into diverse environments, including cloud platforms and enterprise systems. Through its extensibility, PostgreSQL can support replication, backup, and distributed database solutions, making it suitable for both small systems and large-scale distributed applications [31].

In this thesis, PostgreSQL was used as the primary database management system. Integration with Java through Spring Boot and Hibernate enabled efficient CRUD operations and allowed developers to focus on business logic while minimizing complex SQL code.

This chapter examined the principal methods and tools underlying modern passwordless authentication systems. Solutions based on asymmetric cryptography provide both enhanced security and improved usability by eliminating passwords. The implementation of such systems relies on effective use of standards such as FIDO2, WebAuthn, and CTAP, as well as integration of modern hardware and software components, including YubiKey tokens and biometric authenticators. Furthermore, development tools such as Java, Spring Boot, Hibernate, and Maven form a robust and scalable platform for building high-performance passwordless authentication systems.

3 IMPLEMENTATIONS OF A PASSWORDLESS AUTHENTICATION SYSTEM

This chapter provides a detailed description of the development process of a WebAuthn-based service, including software requirements analysis, system architecture, database schema implementation, development tools, service testing and configuration, hardware requirements, and security assurance.

3.1 Software Requirements Analysis

The service must provide the following functionality:

- user registration via WebAuthn;
- authentication via WebAuthn;
- support for hardware tokens (YubiKey, Google Titan Security Key, Kensington VeriMark, and others) used for user verification and private key storage;
- support for biometric authentication methods (fingerprints, facial recognition);
- generation and processing of authentication tokens for managing user sessions after successful authentication;
- verification of user authenticity based on received data (public key and signature).

Authenticated users must be able to:

- update their profile information;
- manage authentication keys;
- log out and remove session cookies.

3.2 Software Architecture

Software architecture defines the structure and organization of system components. It includes a detailed description of system elements, their interactions, and the principles governing their design and evolution. The architectural foundation

is based on the division into functional modules with clearly defined roles, ensuring modularity, scalability, and component reusability.

3.2.1 Component Models and Interactions

Component models and their interactions represent a critical aspect of software architecture, enabling understanding of how system elements cooperate.

In the architecture of a passwordless authentication system, several key components are distinguished, each performing specific functions and interacting with others through well-defined interfaces. The primary interacting components are the user, client (browser), web server, and authenticator.

The user is the individual who initiates the authentication process to gain access to protected system resources [32]. In passwordless authentication, the user plays a central role by confirming access through physical or biometric factors.

The client/browser serves as the interface between the user and the web system [33]. The browser sends requests to the authentication server, receives challenges, and forwards authenticator responses back to the server. WebAuthn support enables interaction with hardware tokens such as YubiKey and ensures secure transmission of cryptographic data required for modern authentication mechanisms.

The web server receives and processes client requests, manages the authentication workflow, interacts with the database to validate user accounts and token authenticity, and integrates with authenticators using FIDO2/WebAuthn standards [34].

The authenticator is a hardware or software component responsible for generating and storing cryptographic keys used in authentication. Hardware tokens such as YubiKey or Google Titan Security Key enhance security and enable passwordless authentication [35]. The authenticator produces a signature using the private key to confirm user identity and transmits it to the server via the browser [36].

Figure 3.1 illustrates the overall architecture and primary interactions between components in a WebAuthn-based system.

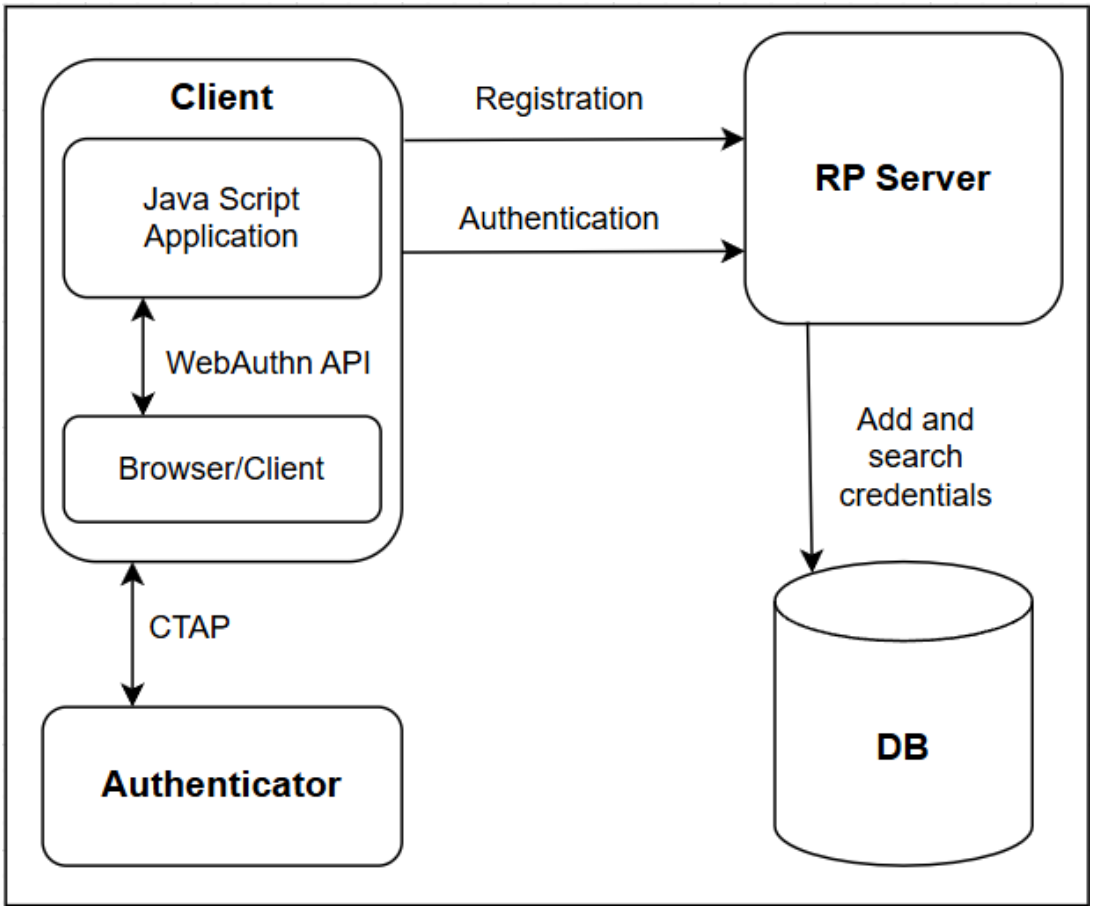

Figure 3.1 – Overall architecture of component interaction in a WebAuthn-based system

The diagram in Figure 3.1 includes the following interactions:

- the client (browser) initiates registration and authentication requests via the WebAuthn API by interacting with a JavaScript application;

- the authenticator communicates with the browser through the CTAP protocol, providing secure cryptographic operations;

- the web server is responsible for processing requests and transferring data to the database, where user accounts, public keys, and other related information are stored.

Figure 3.2 presents a sequence diagram illustrating the interaction of these components.

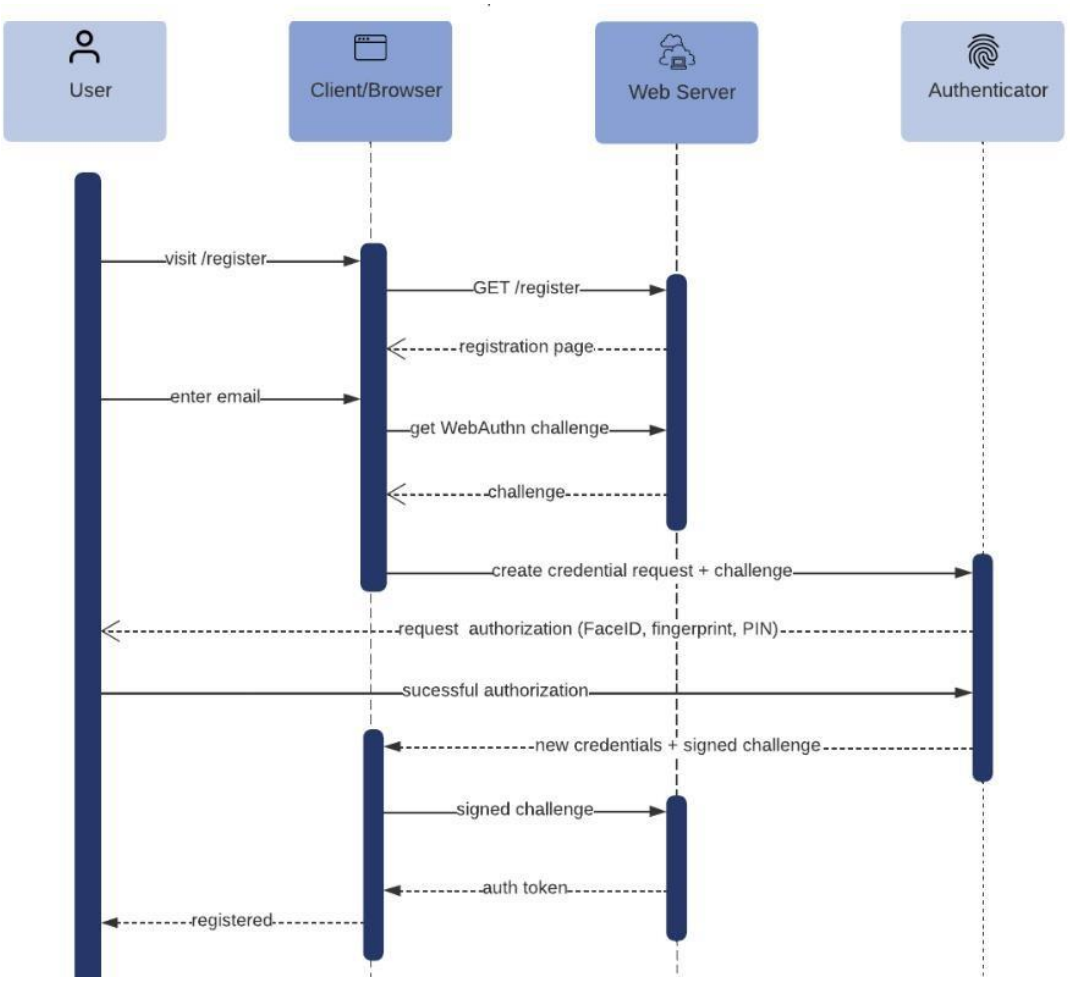

Figure 3.2 – Sequence diagram of component interactions

The diagram presented in Figure 3.2 consists of eight steps:

1. The user navigates to the registration page (visit /register) via the client/browser. The client sends a GET request to the server to retrieve the registration page.

2. The user enters their email address (enter email), and the client transmits this data to the server.

3. Upon receiving the request, the server generates a WebAuthn challenge and sends it to the client/browser for further authentication.

4. The client creates a credential creation request using the received challenge and forwards it to the authenticator. The authenticator may be a YubiKey, Face ID, or another hardware or biometric device.

5. The user completes verification using biometric data (fingerprint, Face ID) or a PIN code. After successful verification, the authenticator signs the challenge with

the private key.

6. Following successful authentication, the authenticator returns the newly created credentials and the signed challenge to the client.

7. The client/browser transmits these credentials and the signed challenge to the server. The server verifies the signature and, if valid, generates an authentication token (auth_token) for the user.

8. The user receives confirmation that registration has been completed successfully and that they are now registered in the system.

These components communicate with each other through APIs that must be designed to ensure secure and efficient interaction. Equally important is the development of an administrative panel that enables system management, configuration of security parameters, and control over user accounts and access permissions.

Defining an appropriate architecture is critical to the success of any software development project. It must account not only for current requirements but also for future scalability and system evolution.

3.2.2 Server-Side Implementation

The implementation of the server-side component of a passwordless authentication system requires clear code structuring to support scalability and maintain security. In this context, the experimental system was organized into packages, each performing specific functions, thereby promoting modularity and simplifying both development and maintenance. The package structure of the server-side component is illustrated in Figure 3.

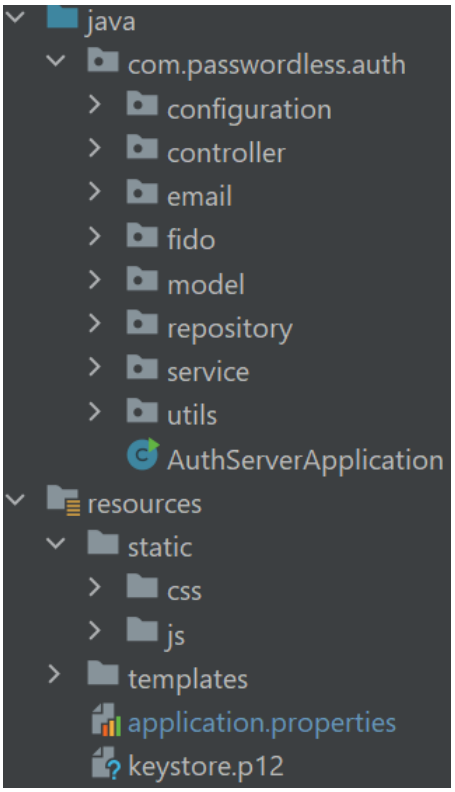

Figure 3.3 – Server-side package structure

The structure presented in Figure 3.3 includes the following packages:

- the com.passwordless.auth.configuration package is responsible for configuring security and authentication within the system using Spring Security. Its core components include WebSecurityConfig, which defines access to resources, as well as filters for token- and cookie-based authentication. In addition, this package contains WebAuthn integration through RelyingPartyConfiguration for working with Yubico;

- the com.passwordless.auth.controller package implements APIs for managing user authentication and registration via WebAuthn. It processes HTTP requests for login, logout, registration, and account recovery, ensuring integration with client applications and authentication services;

- the com.passwordless.auth.email package implements functionality for sending email messages required for account recovery in case of device loss;

- the com.passwordless.auth.fido package implements FIDO2/WebAuthn authentication mechanisms. It contains classes for processing FIDO tokens, converting HTTP requests into authentication objects, managing verification through FidoAuthenticationManager, and handling successful authentication via FidoLoginSuccessHandler. This package integrates WebAuthn with the Spring

Security framework, enabling secure passwordless authentication;

- the com.passwordless.auth.model package contains data models used for storing system information. It includes classes representing roles, authentication keys, users, access tokens, and login and registration requests;

- the com.passwordless.auth.repository package contains repositories for database access, providing storage and management of user accounts and requests related to registration and authentication;

- the com.passwordless.auth.service package contains services that implement business logic for authentication processing, user registration, and account management;

- the com.passwordless.auth.utils package contains utility classes that provide auxiliary functionality for working with JSON, URLs, tokens, user accounts, and UUIDs.

The resources directory includes two subdirectories for implementing the client side: static, which stores JavaScript and CSS files, and templates, intended for storing HTML files.

For a more detailed understanding of component interaction within the experimental system, Figure 3.4 presents a class diagram illustrating their structure and relationships.

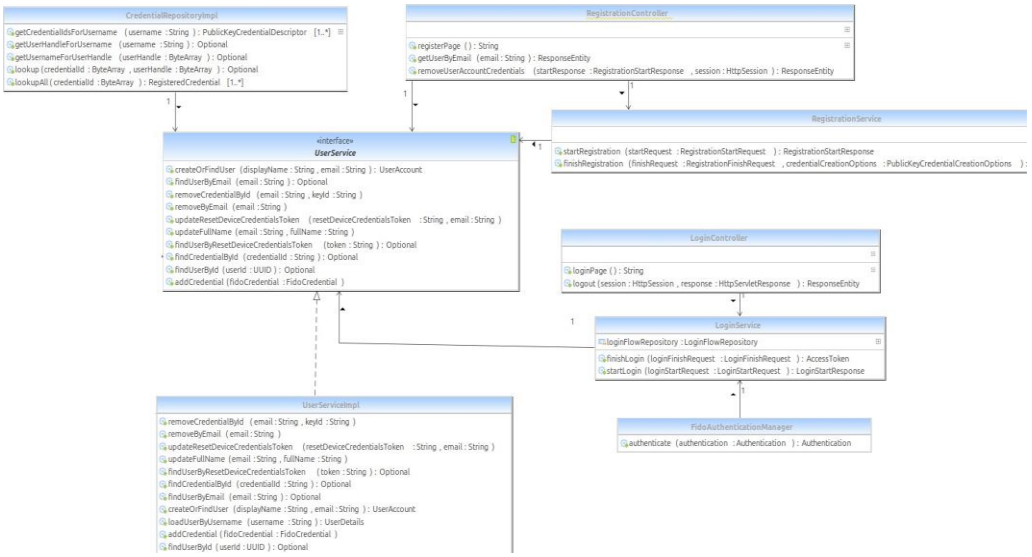

Figure 3.4 – Class diagram of the experimental system

Figure 3.4 presents the architecture of the core system components that support passwordless authentication using the WebAuthn protocol. It clearly illustrates the

relationships between controllers, services, and repositories, which together form the foundation for request handling, business logic execution, and database access.

The controllers (LoginController, RegistrationController, UserAccountController) act as entry points for client interaction with the system. They process requests such as new user registration, system login, and account management, forwarding them to the appropriate services.

The services (LoginService, RegistrationService, UserServiceImpl) are responsible for implementing the core functionality. LoginService performs user authentication, including validation of FIDO2 tokens and generation of access tokens. RegistrationService implements the registration workflow by creating public authentication keys and associating them with users. UserServiceImpl manages user accounts, enabling creation, update, and deletion of user data, as well as management of public keys stored in the server database.

The repositories (UserAccountRepository, LoginFlowRepository, RegistrationFlowRepository, CredentialRepositoryImpl) provide access to the database and manage data persistence. They ensure efficient storage and retrieval of user information and authentication data.

3.2.3 Database Schema Implementation

The service processes requests for passwordless registration and authentication. A relational PostgreSQL database is used for data storage.

A relational database is a structured collection of data organized into predefined relations, where information is stored in one or more tables consisting of rows and columns [37].

For the implementation of this service, a set of database tables was created, the structure of which is shown in Figure 3.5. The schema includes seven tables with their respective fields and data types.

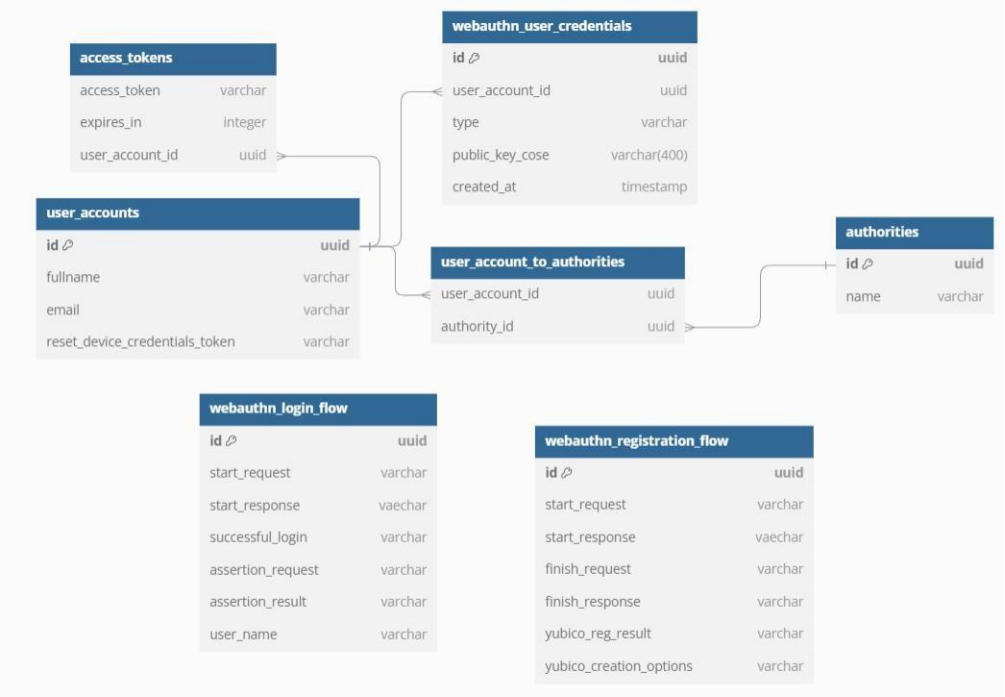

Figure 3.5 – Database Schema

The `user_accounts` table stores the core information about users, which is fundamental for identification and authentication. It contains such fields as the unique user identifier (user_id), full name, email address, and a device credential reset token. This token is used to restore access or reconfigure the authenticator, which is essential for maintaining both system security and user convenience.

The `authorities` table contains roles or access privileges, while the `user_account_to_authorities` table links users to their respective roles via foreign keys.

The `webauthn_user_credentials` table stores critically important data required for passwordless authentication, including the public key, the credential type (e.g., platform or roaming authenticator), and additional metadata associated with the authenticator. This ensures the uniqueness of each user's authentication method and significantly enhances the overall security of the system.

The `webauthn_login_flow` table maintains a detailed log of the user login process. It records all requests and responses exchanged during authentication, including the start and end timestamps, operation status, and possible errors. This enables system administrators to track login history, analyze potential failures, and maintain effective security monitoring.

The `webauthn_registration_flow` table stores information about the user

registration process, including the initial and final requests, as well as the results of registration performed via Yubico services.

The `access_tokens` table is responsible for storing access tokens, including their expiration time and a reference to the user to whom each token belongs.

3.3 Implementation of the Registration and Login Interface

Before accessing the system, a user must complete registration on the page located at "/auth/webauthn/register", which is shown in Figure 3.6.

In addition, the user is required to fill in two mandatory fields: first and last name, and email address.

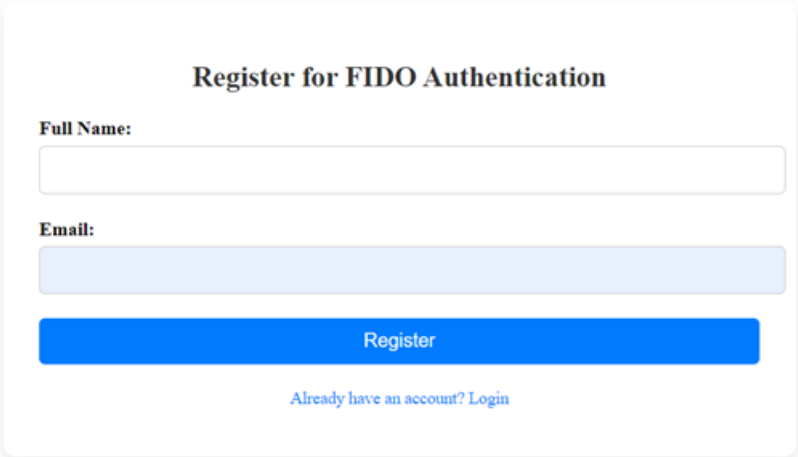

Figure 3.6 – System Registration Form

After clicking the "Register" button, a pop-up window appears for the user, offering a location for storing the passkey. In general, there are three different options:

1. Google Password Manager allows storing keys in Google's cloud storage, enabling users to sign in from multiple devices connected to the same Google account. This option is less secure compared to others due to cloud-based storage, but it is very convenient for users who work with several devices.

2. Windows Hello or an external hardware key ensures that the key is stored locally on the device or on an external hardware token (for example, YubiKey). This method is considered significantly more secure, as the key is never transmitted over

the network and is stored physically on the device or token, access to which is available only to the user.

3. Using another phone, tablet, or hardware key. This method allows the key to be stored on a different device that can later be used for authentication. For example, a user may use a smartphone as an authenticator to sign in on a computer. This provides additional flexibility for those who use multiple devices but prefer to keep the key separate from the primary one.

The selection among these authenticators is shown in Figure 3.7, where the user is offered to choose the most convenient option for storing passkeys.

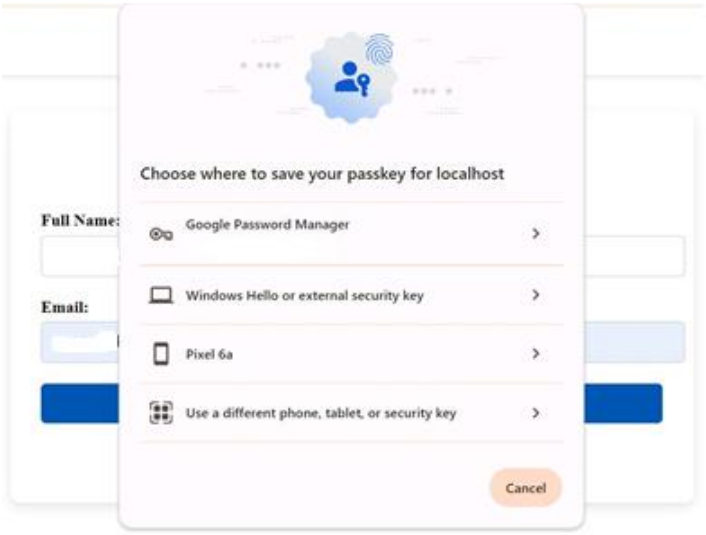

Figure 3.7 – Pop-up Window for Selecting an Authenticator

Next, depending on the user’s choice, a different pop-up window will appear.

If the user selects Google Password Manager and chooses to store the passkey in Google’s cloud, a pop-up window will appear requesting confirmation of signing in to the Google account. If the user is already logged into their Google account, the system will automatically associate the new passkey with that account.

In the same window, shown in Figure 3.8, a message appears indicating that the passkey will be stored in Google’s cloud storage and will be available for signing in from any device linked to the account. To complete the registration process, the user must click “Create.”

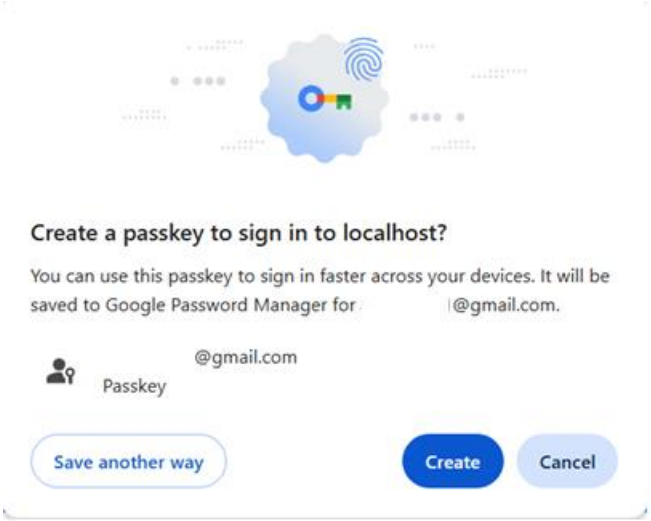

Figure 3.8 – Notification about Saving the Key in Google Cloud

If the user selects Windows Hello, a dialog window with biometric verification or a PIN prompt appears. To successfully register the key, the user must confirm their identity using the selected method: fingerprint scanning, facial recognition, or PIN entry. An example is shown in Figure 3.9.

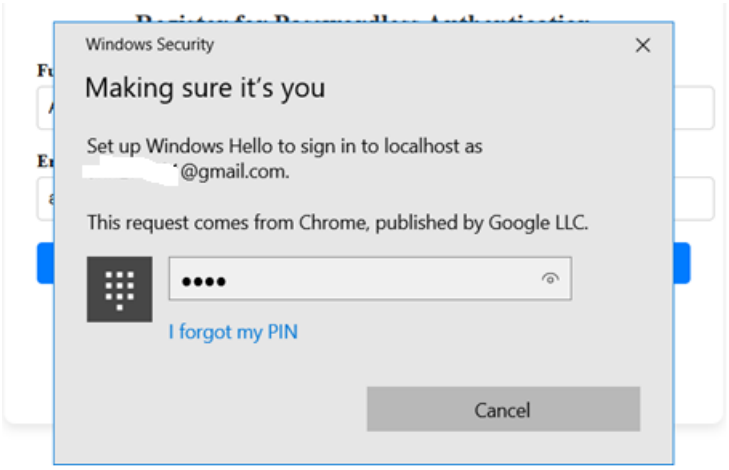

Figure 3.9 – Windows Hello PIN Entry Prompt

If an external hardware key, such as a YubiKey, is selected, a window appears as shown in Figure 3.10 with instructions to insert the key into a USB port or bring it close to a device that supports NFC. This message guides the user through the necessary steps to configure the key.

When using an NFC connection, the user only needs to bring the key close to a device equipped with an NFC module, which makes the process convenient for

mobile devices. If the key is connected via USB, the user must insert it into the appropriate port.

Additionally, if required by the hardware key configuration, the user must touch the sensor on the key to activate it and enter a PIN code or password, which adds an extra layer of security.

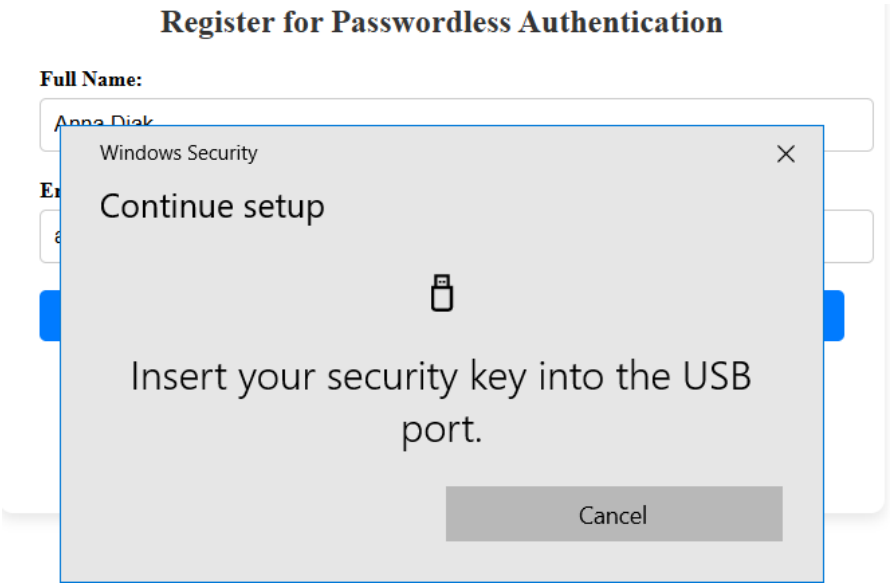

Figure 3.10 – Prompt to Insert the Key into the USB Port

If the user chooses to use another device, the system displays the window shown in Figure 3.11, prompting the user to scan a QR code using another device (a phone or tablet) on which the key will be stored. The QR code is generated by the system to identify the user's account and transmit the required information to the other device.

After scanning the QR code on the additional device, a confirmation window for key creation is displayed. This approach provides convenience and flexibility for users who prefer to store their authentication keys on separate hardware.

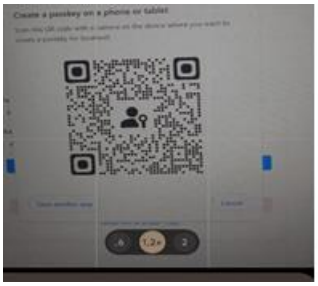

Figure 3.11 – Prompt to Scan the QR Code

After scanning the QR code on the other device, a confirmation window for key creation appears, as shown in Figure 3.12.

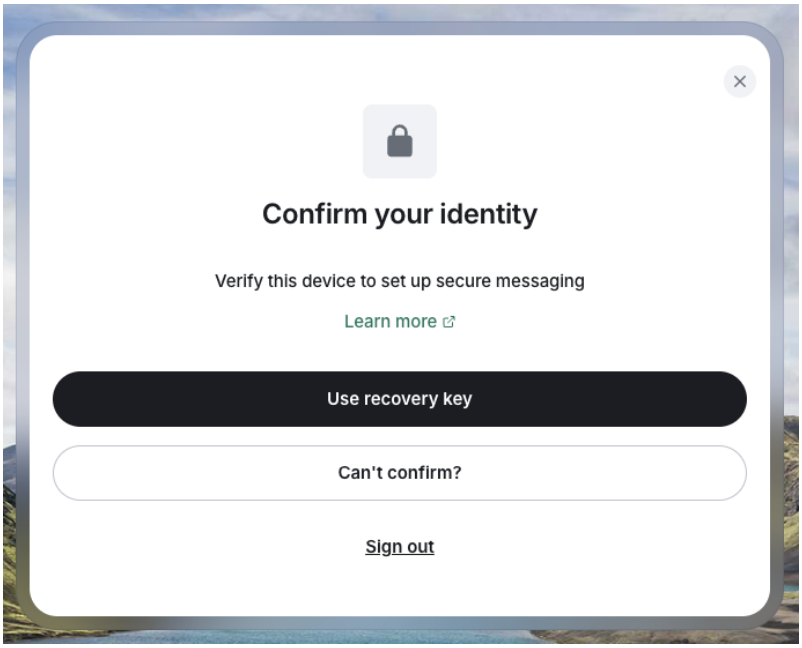

Figure 3.12 – Confirmation Message for Key Creation on the Device

When the user agrees to store the key on the device, they will also be prompted to provide their biometric data in order to confirm the action.

After successful registration in the system using any of the available methods, the user is shown a message indicating that the registration process has been completed, as illustrated in Figure 3.13.

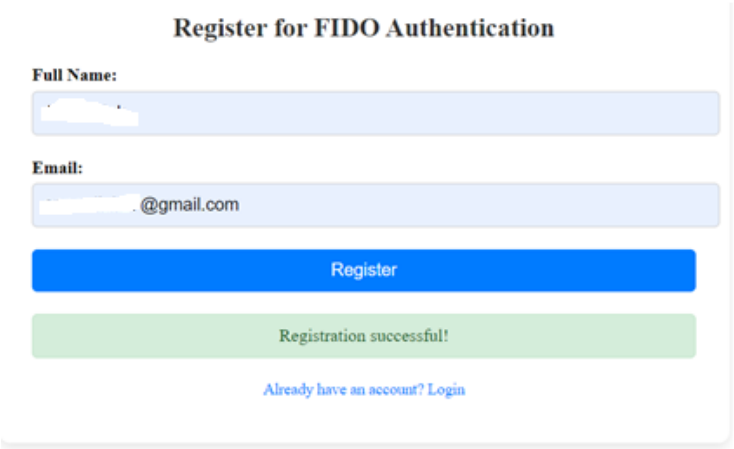

Figure 3.13 – Display of the Form After Successful Registration

It should also be noted that the user is given 30 seconds to create the keys. If the user fails to complete the process within the specified time or cancels the registration, a corresponding notification is displayed, as shown in Figure 3.14.

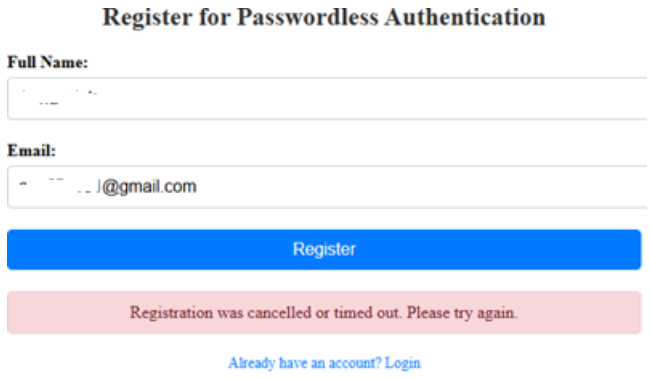

Figure 3.14 – Error Message During the Registration Process in the System

After completing the registration, the user must proceed to the login page located at “/auth/webauthn/login”. On this page, the user needs to enter their email address and select the authenticator that was previously used during registration. The login form is shown in Figure 3.15.

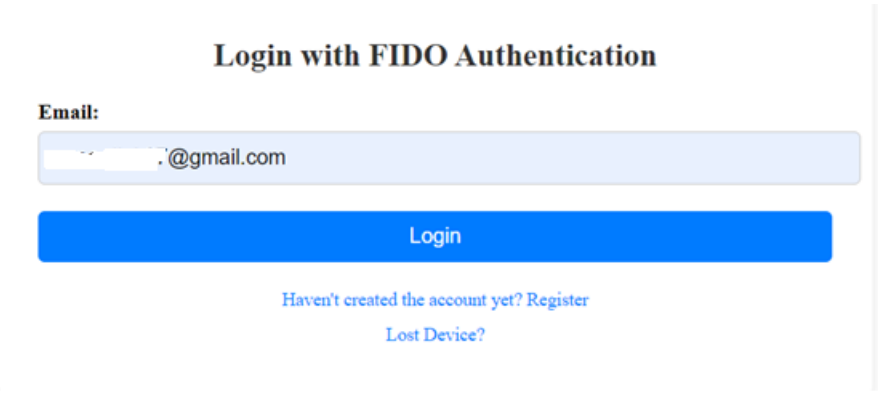

Figure 3.15 – System Login Page

When authentication is completed successfully, the user is redirected to their personal account dashboard.

In the event that a device is lost, access to the account can be restored via the previously registered email address. To do this, the user must navigate to the page "/auth/webauthn/lost_device", shown in Figure 3.16, and enter their email address.

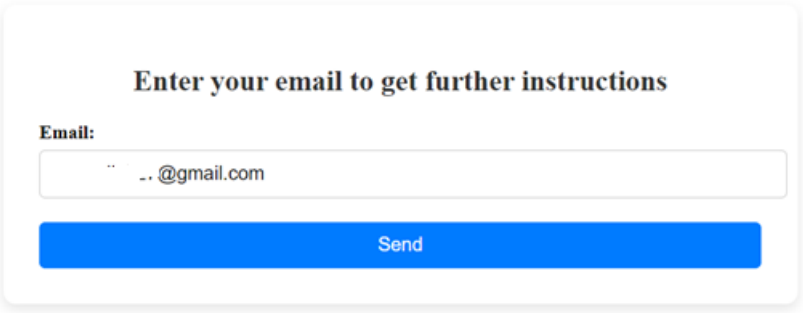

Figure 3.16 – Page for Restoring Account Access from a New Device via Email

After the instructions are successfully sent, they will appear in the previously provided email inbox. The instructions for adding new keys are shown in Figure 3.17.

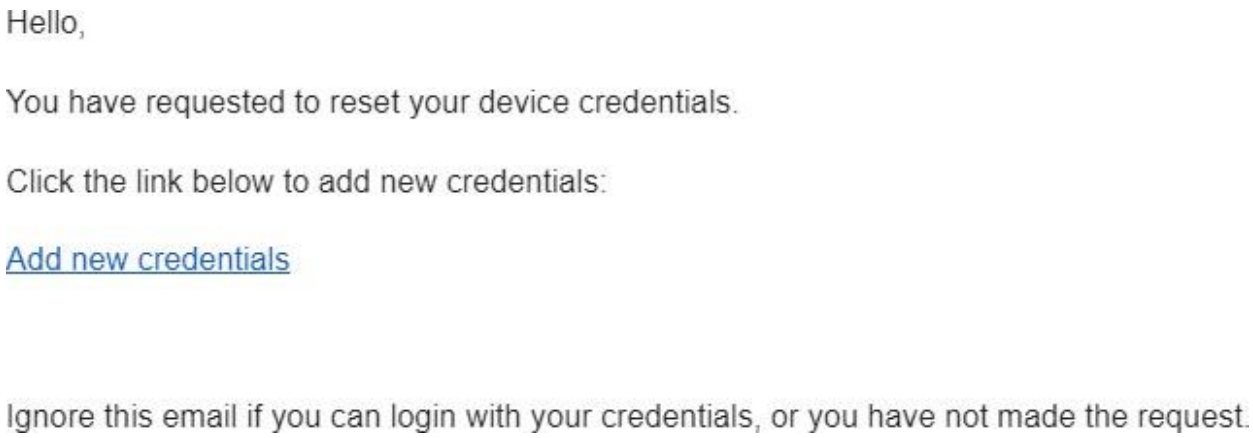

Figure 3.17 – Instructions for Gaining Access from a New Device

After following the link, the user will see the form shown in Figure 3.18.

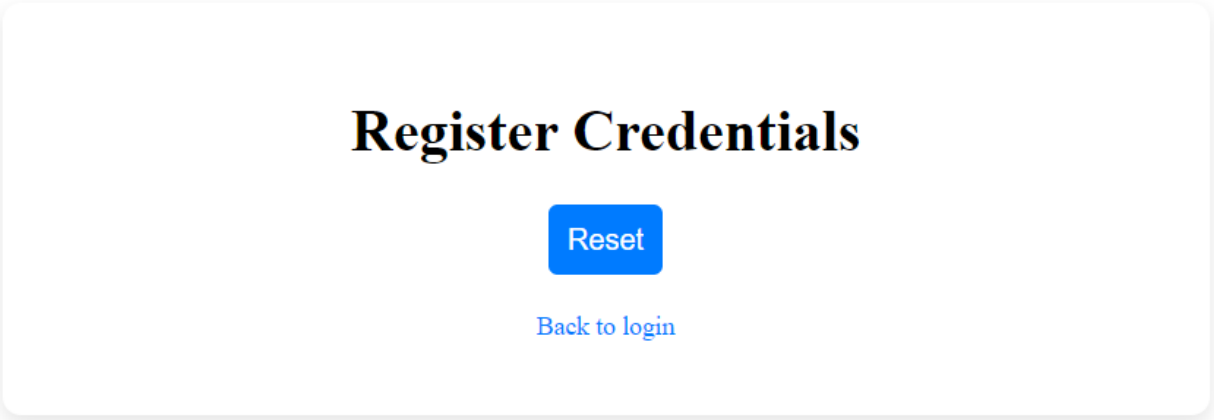

Figure 3.18 – New Key Registration Form

To gain access to the account from a new device, the user must click the "Reset" button. As a result, the same window shown in Figure 3.7 is displayed, offering the option to select an authenticator.

Based on the figures presented in this section, which illustrate the processes of registration and login in the system, it is possible to observe how the client-side component interacts with the server to perform authentication operations. These processes ensure secure data exchange, including account verification, generation of WebAuthn challenges, and management of authentication keys.

To provide this functionality, the system uses APIs responsible for the different stages of registration, login, and access recovery. Table 3.1 below presents a detailed description of these requests, which implement the integration between the client side and the server.

Table 3.1 – Main API Endpoints for Client–Server Interaction

No.	Request	Description	Input Data	Output Data	Usage
1.	GET /auth/webauthn/register/user	Checks whether a user with the specified email exists	email	User object or null	Before starting registration
2.	POST /auth/webauthn/register/start	Starts the registration process and generates a challenge	JSON: { fullName, email }	Object containing the challenge and related data	To initiate registration
3.	POST /auth/webauthn/register/finish	Completes registration and stores the public key	JSON: { flowId, credential: { id, response } }	Registration completion status	To finalize registration
4.	POST /auth/webauthn/login/start	Starts the login process and generates a challenge	JSON: { email }	Object with authentication data	To initiate login

End of Table 3.1.

5.	POST /auth/webauthn/login/finish	Completes the login process and verifies the public key	JSON: { flowId, credential: { id, response } }	Access token (JWT)	To finalize authentication
6.	POST /auth/webauthn/lost_device	Sends an email for access recovery	JSON: { email }	Status: success, not found, error	When the authenticator device is lost

These API endpoints provide all the necessary operations for user registration, login, and access recovery by integrating the client-side component with the server-side logic of the system. They clearly structure the authentication workflows and ensure correct interaction within the implemented functionality.

3.4 Implementation of the User Account Dashboard

After a successful login, the user is redirected to a predefined endpoint specified by the system configuration. In the test system, the user can access their account via the route /auth/account, where functionality for editing personal data (except for the email address) is provided. This feature is illustrated in Figure 3.19.

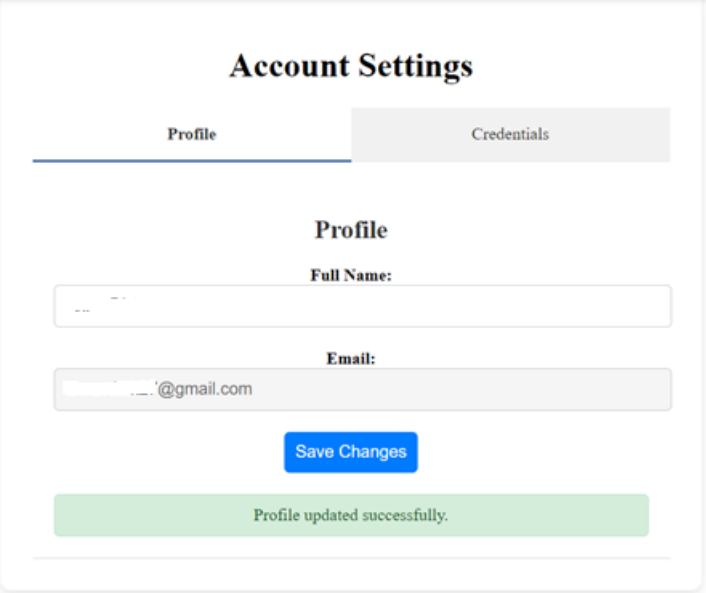

Figure 3.19 – Form for registering new keys

The user can also view, add, and delete registered keys associated with their account across different devices. This provides flexibility in managing authenticators, allowing users to keep their keys up to date, remove those that are no longer used, or add new ones, for example, when switching to another device.

Keys contain important information used for secure authentication. In particular, each key includes the following fields:

- Creation date – helps track when the key was created, which is useful for managing key lifetimes or identifying obsolete entries;

- Key ID – a unique identifier that allows the system to distinguish the key among others associated with the account;

- Public Key – the public key used to verify signatures during authentication. This key is part of the key pair and does not require secure storage, as it does not disclose private information.

The private key, which is the foundation of system security, remains stored only on the device on which it was created. For security reasons, information about the exact storage location of the private key is not displayed in the system. This prevents potential data leaks and ensures that the private key remains accessible only on the originating device.

Users manage these keys through a dedicated interface in the account settings, shown in Figure 3.20. For example, when a user deletes a key, the system guarantees that this key can no longer be used for authentication. Similarly, when adding a new

key, the user is prompted to create a new key pair in compliance with all required security measures.

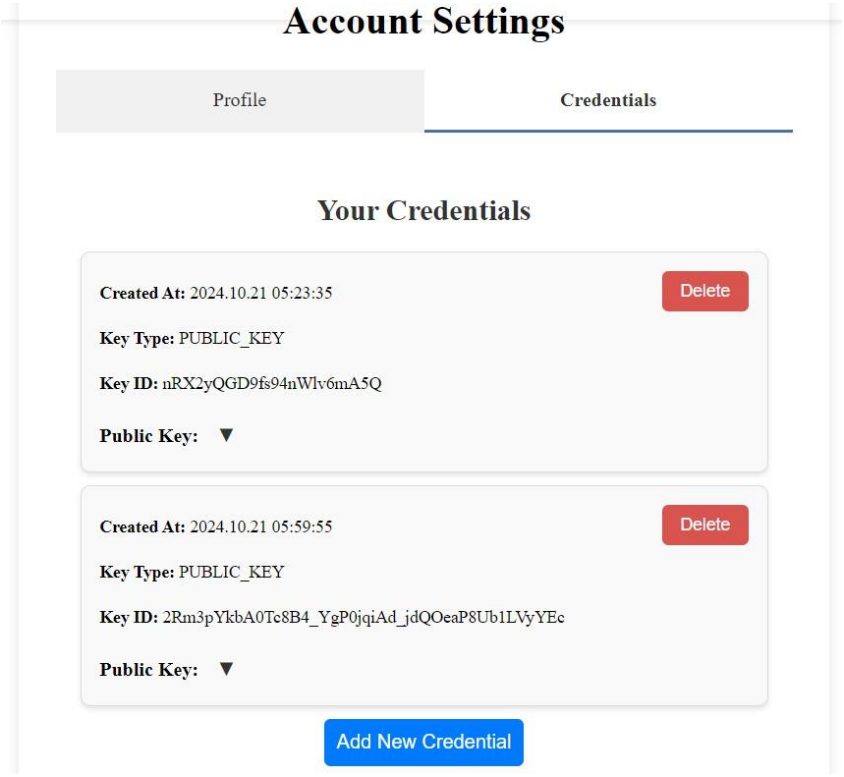

Figure 3.20 – List of keys registered in the user account

This approach is convenient in cases of device replacement or loss of an authentication device.

3.5 Service Testing and Configuration

To configure and test the test system, it is necessary to ensure the correct configuration of the environment variables used to run the application. All configurations are performed using the `run_script.sh` script, located in the project's root directory. Figure 3.21 presents this script, demonstrating the key variables for the system's operation. These variables include SSL configuration, database connection parameters, and post-login redirection settings.

```
#!/bin/bash

# environment variables
export SERVER_PORT=2024

export SERVER_SSL_ENABLED=true
export SERVER_SSL_KEY_STORE_TYPE=PKCS12
export SERVER_SSL_KEY_STORE=classpath:/keystore.p12
export SERVER_SSL_KEY_STORE_PASSWORD=changeit
export SERVER_SSL_KEY_ALIAS=demo

export SPRING_DATASOURCE_URL=jdbc:postgresql://localhost:5432/as
export SPRING_DATASOURCE_USERNAME=postgres
export SPRING_DATASOURCE_PASSWORD=postgres
export SPRING_DATASOURCE_DRIVER_CLASS_NAME=org.postgresql.Driver

export SERVER_REDIRECT_URI=/auth/account

 # Replace with your project's directory
PROJECT_DIR="D:\Labs\Diploma 2\authorization-server"
cd "$PROJECT_DIR" || exit

# Run the Spring Boot application
echo "Running the Spring Boot application..."
java -jar target/*.jar \
  --server.port=$SERVER_PORT \
  --server.ssl.enabled=$SERVER_SSL_ENABLED \
  --server.ssl.key-store-type=$SERVER_SSL_KEY_STORE_TYPE \
  --server.ssl.key-store=$SERVER_SSL_KEY_STORE \
  --server.ssl.key-store-password=$SERVER_SSL_KEY_STORE_PASSWORD \
  --server.ssl.key-alias=$SERVER_SSL_KEY_ALIAS \
  --spring.datasource.url=$SPRING_DATASOURCE_URL \
  --spring.datasource.username=$SPRING_DATASOURCE_USERNAME \
  --spring.datasource.password=$SPRING_DATASOURCE_PASSWORD \
  --spring.datasource.driver-class-name=$SPRING_DATASOURCE_DRIVER_CLASS_NAME \
  --server.redirect-uri=$SERVER_REDIRECT_URI
```

Figure 3.21 – Configuration and startup script of the test system in run_script.sh

SSL configuration is defined using the variables in Table 3.2.

Table 3.2 – Variables for SSL configuration

Variable	Description
SERVER_SSL_ENABLED	Determines whether the server operates via HTTPS (must always be true)

End of Table 3.2

SERVER_SSL_KEY_STORE_TYPE	Format of the keystore file containing certificates
SERVER_SSL_KEY_STORE	Path to the keystore file containing SSL certificates
SERVER_SSL_KEY_STORE_PASSWORD	Password for accessing the keystore file
SERVER_SSL_KEY_ALIAS	Alias for selecting a specific key in the keystore

Without these parameters, WebAuthn service operation is impossible, as the standard requires a secure HTTPS connection.

To enable interaction with the database, the variables shown in Table 3.3 must be configured.

Table 3.3 – Database configuration variables

Variable	Description
SPRING_DATASOURCE_URL	URL for database connection, including server, port, and database name
SPRING_DATASOURCE_USERNAME	Username for database access
SPRING_DATASOURCE_PASSWORD	Password for database access
SPRING_DATASOURCE_DRIVER_CLASS_NAME	Driver class for database connection

These parameters allow the system to store and retrieve user data, WebAuthn

keys, and other objects.

At the end of configuration, it is important to define the variable SERVER_REDIRECT_URI, which specifies the address to which the user will be redirected after a successful login.

The run_script.sh script can be executed directly, which automatically configures all necessary parameters for system startup. This requires executing ./run_script.sh in the terminal. Alternatively, the script can be integrated into Docker for more convenient deployment. In this case, variable configuration is passed to the container via command-line arguments, providing flexible setup for different environments.

System operation can be tested through key endpoints such as user registration, login, and account access. Testing ensures the correctness of configuration and proper functioning of all components, including security and database integration. Figure 3.21 details all key configuration elements and serves as the foundation for correct system operation.

3.6 Hardware Requirements

To ensure stable and reliable operation of the passwordless authentication service based on WebAuthn, certain technical requirements must be met to guarantee efficiency, performance, and security.

The server hosting the service must have a 4-core processor to ensure fast processing of multiple concurrent user requests. This is critical for maintaining high performance under heavy load. The minimum amount of RAM should be 8 GB, allowing the system to retain necessary data in memory and avoid delays. At least 20 GB of free disk space must be provided to store user information, authentication keys, session data, and activity logs.

For the PostgreSQL database used to store authentication keys, session records, and other critical data, a server with a 2-core processor and at least 4 GB of RAM is recommended. Disk space requirements depend on the volume of accumulated data, so additional capacity should be planned to accommodate growth in users and sessions.

For proper WebAuthn operation, users must use modern browsers that support this standard. Table 3.4 presents WebAuthn support information for various browsers based on data from caniuse.com.

Table 3.4 – WebAuthn support across browsers

Browser	Supported Versions (Full)	Partially Supported Versions	Not Supported / No Support
Google Chrome	67 and newer	Not supported	4–66
Mozilla Firefox	60 and newer	2–59 (experimental features required)	Not supported
Microsoft Edge	18 and newer	12–17 (experimental features required)	≤ 12
Apple Safari	13 and newer	3.2–12.1 (experimental features required)	≤ 3.1
Opera	54 and newer	9–53	Not supported
Opera mini	Not supported	Not supported	Not supported

Table 3.4 demonstrates WebAuthn compatibility across popular browsers and provides a clear overview of supported environments, helping users choose an appropriate browser.

Google Chrome fully supports WebAuthn starting from version 67. Versions 4–66 do not support this standard, so users are advised to update their browsers.

Mozilla Firefox provides full support from version 60. Versions 2–59 offer partial support, where certain functions such as external hardware tokens (e.g.,

YubiKey) may work unreliably or require additional configuration.

Microsoft Edge fully supports WebAuthn starting from version 18. Versions 12–17 provide partial support, as WebAuthn was available only in experimental mode. Version 12 does not support WebAuthn.

Apple Safari fully supports WebAuthn from version 13. Versions 3.2–12.1 provide partial support, where WebAuthn is available only as an experimental feature and may require developer settings. Versions 3.1 and earlier do not support WebAuthn.

Opera fully supports WebAuthn starting from version 54. Versions 9–53 provide partial support, where advanced features may not function properly without enabling experimental options.

For effective WebAuthn-based authentication, it is also important to consider authenticator support across operating systems. WebAuthn uses two main types of authenticators: platform and roaming. However, support varies depending on browser, operating system, and connection type (USB, NFC, BLE).

	Android 7+	iOS 14.5+	Windows 10 *(with Windows Hello)*	macOS Catalina	macOS Big Sur	Desktop Linux
Chrome	Yes	Yes	Yes	Yes	Yes	-
Safari	N/A	Yes	N/A	No	Yes	N/A
Firefox	No	Yes	Yes	No	No	-
Brave	No	Yes	Yes	Yes	Yes	-
Edge	No	Yes	Yes	Yes	Yes	-
Internet Explorer	N/A	N/A	No	N/A	N/A	N/A

Figure 3.22 – Compatibility matrix of browsers and operating systems supporting built-in authenticators (Touch ID, Face ID, or Windows Hello)

In Figure 3.22, the notation "Yes" indicates full support for platform authenticators in a specific browser and operating system combination. The "No" mark signifies a lack of support, which may render the authentication functionality unavailable. "N/A" (Not Applicable) means that the specified combination is not applicable or not supported. A hyphen is used to denote a lack of information or uncertain support for a particular combination.

These same notations apply to Figure 3.23, which reflects the compatibility of browsers and operating systems with roaming authenticators, such as YubiKey and Titan Key.

Analyzing the matrix in Figure 3.22, it can be concluded that support for platform authenticators largely depends on the combination of browser and operating system. The best compatibility is observed in modern browsers, such as Google Chrome, Safari, and Microsoft Edge, running on popular operating systems, including Windows 10, macOS Big Sur, Android, and iOS [38].

Conversely, support on older operating systems or less common browsers, such as Firefox on macOS or Desktop Linux, is limited. This underscores the necessity of updating browsers and operating systems to their latest versions to ensure the full functionality of WebAuthn.

Next, we will examine the support for roaming authenticators depending on the browser and OS in Figure 3.23.

	Android 7+	iOS 14.5+	Windows 10 *(with Windows Hello)*	macOS Catalina	macOS Big Sur	Desktop Linux
Chrome	Yes	Yes	Yes	Yes	Yes	-
Safari	N/A	Yes	N/A	No	Yes	N/A
Firefox	No	Yes	Yes	No	No	-
Brave	No	Yes	Yes	Yes	Yes	-
Edge	No	Yes	Yes	Yes	Yes	-
Internet Explorer	N/A	N/A	No	N/A	N/A	N/A

Figure 3.23 – Browser and Operating System Compatibility Matrix Supporting External Hardware Authenticators (YubiKey, Titan Key)

From Figure 3.23, it can be concluded that the most stable support for these devices is provided via USB connectivity. This form of authentication is supported by nearly all modern browsers on popular platforms such as Windows, macOS, Android, and Linux [38].

However, support for NFC and Bluetooth (BLE) is less widespread and depends on the browser and operating system. For example, Safari and Firefox browsers have limitations when working with BLE and NFC, which can affect the

convenience of using external tokens.

Adhering to these technical requirements will help ensure the uninterrupted operation of your service and efficient processing of user requests.

This section provides a comprehensive overview of the procedure for implementing a WebAuthn-based passwordless authentication system. It details the key stages, from software requirement analysis to service testing and configuration. The software architecture was designed to ensure robust interaction between all system components, which include the client, server, and authenticators. A database schema was created for securely storing authentication keys and effectively managing user accounts.

Furthermore, intuitive registration, login, and account management panels were implemented, adhering to modern security standards and offering flexibility in authenticator selection. The interface supports various authenticator types, including platform and roaming authenticators, ensuring a universal and secure user experience [39].

The system underwent rigorous testing, including unit, integration, and load tests. Identified issues were resolved to ensure stable operation in real-world conditions. Specific technical requirements for the deployment environment were established, guaranteeing system stability, performance, and compliance with contemporary security standards.

The results presented in this section illustrate the system's readiness for deployment in a production environment, offering users secure and convenient passwordless authentication while maintaining high reliability and user-friendliness.

4 DEVELOPMENTS OF THE STARTUP PROJECT

This section is dedicated to a detailed examination of the key aspects of developing a startup project. It includes defining the concept, conducting a technological audit, analyzing the market and competitive landscape, assessing risks and opportunities, as well as developing a market entry strategy and marketing program. Particular attention is paid to analyzing the target audience and identifying the project's strengths and weaknesses, which allows for the formation of an effective positioning strategy and ensures the successful introduction of the product to the market.

4.1 General Overview of the Startup Project

Let us examine the characteristics of authentication methods presented in Table 4.1.

Table 4.1 - Description of the Startup Project Idea

Concept	Application Areas	User Benefits
Development of a passwordless authentication system based on WebAuthn and FIDO2 standards	Corporate Information Systems	Increased level of access security to systems, reduced risk of data compromise, elimination of password dependency.
	Web Applications	Provision of fast and secure login for users, compatibility with modern browsers and platforms.

End of Table 4.1

	3. Mobile Applications	Integration of biometric authentication methods (fingerprints, facial recognition), enhancing convenience and security for users.
Integration of Passkey based authentication	1. Cloud Services | Ensuring secure access to cloud resources, reducing data leakage risks through cryptographic methods.	Integration of Passkey
	2. Ecommerce Platforms	Building customer trust through secure, passwordless authentication that minimizes the risk of phishing attacks.
Implementation of Magic Links as an additional authentication method	1. One time or Infrequent Access	A convenient authentication method for users who rarely log in, eliminating the need to remember credentials.

The next step involves evaluating the system's strengths and weaknesses to determine its market implementation potential and competitiveness. The corresponding information is presented in Table 4.2.

Table 4.2 - Identification of Project Idea Characteristics

No.	Economic Characteristics	Concept (Strategy) of Competitors			Weak Sides	Neutral Sides	Strong Sides
		Own Project	Competitor 1	Competitor 2			
1	Execution Form	Asymmetric Cryptography, Passkeys	Passwords with MFA	Hardware Tokens, FIDO2			+
2	Ease of Use	Integration with Biometrics	Dependence on password complexity	Limitations for users without tokens			+
3	Cross-Platform Compatibility	High	Low	High			+
4	Implementation Complexity	Moderate	Low	High	+		

Table 4.2 demonstrates that the proposed passwordless authentication system based on asymmetric cryptography and passkeys holds significant advantages over traditional methods, such as passwords with multi-factor authentication (MFA) and FIDO2 hardware tokens. Specifically, it ensures high ease of use through integration with biometric methods, cross-platform compatibility, and moderate implementation complexity. These characteristics make it an attractive choice for modern information systems aiming to enhance both security levels and user convenience.

4.2 Technology Audit of the Idea

This section conducts an audit of technologies applicable for implementing the passwordless authentication system proposed in this thesis. The considered technologies can be seen in Table 4.3.

Table 4.3 - Technological Feasibility of the Project Idea

Project Idea	Implementation Technologies	Technology Availability	Technology Accessibility
Passwordless Authentication System	Java Programming Language	Available	Free, Accessible
	Spring Framework	Available	Free, Accessible
	Hibernate ORM Framework	Available	Free, Accessible
	PostgreSQL Database	Available	Free, Accessible
	Maven Build Tool	Available	Free, Accessible
	JavaScript Programming Language	Available	Free, Accessible

All proposed technologies, such as Java, Spring Framework, Hibernate, PostgreSQL, Maven, and JavaScript, are open-source, free, and readily available for use. They ensure high development efficiency, reliability, and ease of integration into modern information systems.

Thus, the combination of these technologies creates a powerful toolkit for implementing a passwordless authentication system that meets contemporary security and usability requirements.

4.3 Analysis of Market Opportunities for Launching the Startup Project

Analyzing the market opportunities that can be leveraged during project implementation, as well as the threats that may hinder its realization, allows for planning development directions considering the market environment, the needs of potential clients, and competitors' offerings. Table 4.4 presents a preliminary market characterization for the developed startup project.

Table 4.4 – Characterization of the Passwordless Authentication Systems Market

No.	Market Characteristic	Value
1	Number of Market Players	Several large companies and numerous startups
2	Total Annual Sales Volume	Exact data unavailable. The market is growing annually
3	Market Dynamics	Rapid growth driven by increased focus on cybersecurity
4	Barriers to Entry	Low. Requirement to comply with security standards
5	Presence of Demand	High demand among enterprises
6	Average Industry Profitability	No precise data available. Depends on the business model and operational scale

End of Table 4.4

7	Specific Certification Requirements	Compliance with FIDO2 standards and other industry regulations

According to the analyzed characteristics, the passwordless authentication systems market demonstrates significant potential for the introduction of a new product, driven by rapid demand growth and relatively low barriers to entry. Table 4.5 provides a detailed profile of potential clients.

Table 4.5 – Profile of Potential Clients for the Startup Project

No.	Need Driving the Market	Users (Primary Clients)	Behavioral Differences Among Potential Target User Groups	Key Requirements / Expectations
1	Enhancing Access Security to Systems and Data	Enterprises across various sectors, financial institutions, government organizations	Enterprises aim to protect confidential data. Financial institutions focus on securing transactions. Government organizations require compliance with regulatory standards	High level of security, standards compliance, ease of integration
2	Convenience and Speed of Authentication	End-users, company employees	End-users value simplicity. Company employees require fast access to work resources	Intuitive user interface, minimized authentication time

End of Table 4.5

3	Reducing Password Management Costs	Corporate IT departments	IT departments seek to reduce the support burden	Centralized management capability, reduction in password reset requests

Satisfying the requirements of the main user groups is a key success factor for the startup project. Understanding the specific needs of each target audience will enable the development of a system that meets user expectations and provides competitive advantages in the market. Therefore, the primary requirements are enhanced security, authentication convenience and speed, and cost reduction.

Table 4.6 presents the main threat factors, their essence, and possible company responses.

Table 4.6 – Threat Factors

No.	Factor	Nature of the Threat	Possible Company Response
1	Increasing User Demands	Rising expectations regarding security and authentication convenience	Continuous product improvement, implementation of new features
2	Intensifying Competition	New players entering the market with similar solutions	Development of unique offerings, enhancement of service quality
3	Technological Changes	Rapid development of technologies that may render the product obsolete	Investment in research and development, adaptation to new technologies

End of Table 4.6

4	Regulatory Changes	Introduction of new laws and standards in the field of cybersecurity	Monitoring legislative changes, ensuring product compliance with new requirement
5	Cyber Threats	Increase in the number and complexity of cyber attacks	Strengthening protection measures, conducting regular security audits

Identifying these threats allows the company to develop strategies to overcome them and minimize potential risks.

Table 4.7 presents the main opportunity factors, their nature, and possible company actions to leverage them.

Table 4.7 – Opportunity Factors

No.	Factor	Nature of the Opportunity	Possible Company Response
1	Growing Demand for Secure Solutions	Increased interest in passwordless authentication	Active marketing campaign, expansion of the sales market
2	Partnership with Large Companies	Opportunity to integrate with products from well-known brands	Establishing collaborations, developing joint solutions
3	Feature Expansion	Adding new capabilities to the product	Conducting research, implementing innovations
4	Entry into the International Market	Access to the international market	Adapting the product to the requirements of different markets, localization

End of Table 4.7

5	Attracting Investment	Securing additional resources for development	Preparing investment proposals, participating in startup programs

Leveraging the opportunities outlined in Table 4.7 will contribute to the company's growth and strengthen its market position.

Let us conduct a stepwise (level) analysis of competition in the passwordless authentication systems market. This analysis helps understand the current market state and define strategies for enhancing the company's competitiveness in the passwordless authentication sector. The results are presented in Table 4.8.

Table 4.8 – Stepwise (Level) Analysis of Competition in the Passwordless Authentication Systems Market

№	Feature of the Competitive Environment	Manifestation of the Characteristic	Impact on Enterprise Activity (Possible Company Actions to Ensure Competitiveness)
1	Type of Competition – Monopolistic	Presence of several large players and numerous startups	Conducting market research, identifying unique product advantages, developing a strategy
2	Level of Competition – International	Presence of local and international companies	Analyzing global trends, adapting the product to international standards, expanding into new markets

End of Table 4.8

3	Industry Characteristic – Cross-industry	Use of technologies across various sectors (finance, healthcare, IT)	Developing universal solutions capable of meeting the needs of different industries; ensuring product flexibility
4	Competition by Product Types – Product-variety	Diversity of solutions with similar functionality	Enhancing product quality and security, implementing innovative features, improving user experience
5	Nature of Competitive Advantages – Non-price	Competition based on quality, security, and convenience	Investing in research and development, ensuring a high level of customer service, building a strong reputation
6	Competition Intensity – High	Rapid technology development and emergence of new players	Constant market monitoring, quick adaptation to changes, implementation of new technologies and solutions

Research on the competitive environment of passwordless authentication systems indicates the presence of monopolistic competition, where large international corporations coexist with numerous startups. High competition intensity and rapid technological development compel companies to constantly monitor the market, quickly adapt to changes, and implement cutting-edge solutions to remain competitive.

The next step involves conducting an analysis of the competitive environment using Michael Porter's Five Forces model, the results of which are presented in Table 4.9.

Table 4.9 – Analysis of Competition in the Industry According to M. Porter

	Direct Competitors in the Industry	Potential Competitors	Suppliers	Customers	Analysis Component
Analysis Component	Presence of several major players, such as Google, Apple, and Microsoft, who offer their own passwordless authentication solutions	Possibility of new companies, particularly startups developing innovative cybersecurity solutions, entering the market	Absence of critical suppliers, as most technologies are open-source or accessible for developing proprietary solutions	High demand for secure and convenient authentication methods from businesses and end-users. Customers have a choice	Traditional authentication methods, such as passwords and two-factor authentication, can serve as substitutes but are gradually losing popularity due to lower security levels
Conclusions	Moderate intensity from large technology	Low barriers to entry facilitate the emergence	Absence of dependence on specific suppliers	High demand for passwordless solutions	Traditional authentication methods are losing

	companies and potential new market players	of new market participants	reduces risks	increases the importance of meeting customer needs	relevance, which promotes the adoption of passwordless technologies

Based on the conducted competition analysis (Table 4.2), characteristics of the startup project idea (Table 4.5), profile and requirements of potential clients (Table 4.6), as well as factors of the market environment (Tables 4.7 and 4.8), a list of competitiveness factors has been formulated and justified, as presented in Table 4.10.

Table 4.10 – Justification of Competitiveness Factors

No.	Competitiveness Factor	Justification (Factors making the factor significant for comparing competing projects)
1	Universality	The passwordless authentication solution can be integrated into various systems and platforms
2	Ease of Use	An intuitive interface and simple configuration allow users to quickly adapt to our product without requiring special knowledge or extensive training
3	Security Level	The use of modern security standards, such as FIDO2 and WebAuthn, ensures reliable protection against phishing attacks and other threats, increasing user trust in our product
4	Cross-Platform Compatibility	Support for various operating systems and devices allows users to utilize our solution on any platform, enhancing its appeal and convenience

5	Scalability	Our solution can be easily adapted to the needs of both small businesses and large corporations, enabling effective service to different market segments and ensuring stable growth

The competitiveness analysis of our own project has revealed key advantages, such as universality, ease of use, high security level, cross-platform compatibility, and scalability. These characteristics allow our solution not only to compete with existing alternatives but also to provide exceptional value to users, increasing their trust and satisfaction. Consequently, the product is capable of meeting the needs of both individual users and large organizations seeking modern authentication technologies.

To gain a more detailed understanding of competitive advantages, a comparative analysis of strengths and weaknesses has been conducted, considering the aforementioned factors. This analysis includes an assessment of potential risks that could affect market perception of the product, as well as the identification of opportunities for further improvement. The results of the comparative analysis are presented in Table 4.11, which allows for the visualization of key competitiveness aspects and supports effective strategic planning.

Table 4.11 – Strengths and Weaknesses of the Startup Project

No.	Competitiveness Factor	Score 1-20	Competitor Product Rating Scale (-3 to +3)						
			-3	-2	-1	0	1	2	3
1.	Universality	20				+			
2.	Ease of Use	16					+		
3.	Security Level	18			+				
4.	Cross-Platform Compatibility	15		+					
5.	Scalability	17					+		

Next, let's examine Table 4.12, which presents the results of the SWOT analysis.

Table 4.12 – SWOT Analysis of the Startup Project

Strengths	Weaknesses
Innovative passwordless authentication algorithm, providing a high level of security and convenience for users.	Lack of brand recognition in the market, which may complicate attracting new clients.
Cross-platform compatibility of the solution, allowing its integration into various systems and platforms.	Limited financial and human resources, which may affect the speed of development and product deployment.
Ease of use, lowering the entry barrier for new users.	Absence of large-scale marketing campaigns.

The SWOT analysis revealed that our startup project possesses significant strengths, such as an innovative algorithm, cross-platform compatibility, and ease of use. However, to successfully enter the market, it is necessary to overcome weaknesses, particularly by increasing brand recognition and expanding financial and human resources. The rapid development of the industry and growing demand for passwordless solutions present significant growth opportunities. Nevertheless, competition from more established companies and rapid technological progress pose potential threats that should be considered when developing the growth strategy.

Next, in Table 4.13, we will examine project implementation alternatives based on the SWOT analysis.

Table 4.13 – Market Implementation Alternatives for the Startup Project

No.	Market Behavior Alternative	Probability of Securing Resources	Implementation Timelin

1	Integration with popular platforms (e.g., Microsoft Azure Active Directory)	80%	12 months
2	Collaboration with financial institutions (banks or other financial organizations)	60%	15 months
3	Participation in international exhibitions and conferences	50%	6 months

At this stage, a comprehensive market and product analysis has been conducted. Specifically, a detailed competitive analysis has been performed, key market factors have been identified, and their favorability for our startup project has been assessed. An in-depth study of the concept and characteristics of the proposed product has also been carried out. Based on the obtained data, it can be concluded that current market conditions are favorable for the successful market entry of our product. This opens significant opportunities for the further development and scaling of the project.

4.4 Development of the Market Entry Strategy

To effectively implement the passwordless authentication solution, it is necessary to identify the main groups of potential consumers, assess their readiness to adopt the product, estimate potential demand, the level of competition, and the complexity of entering the corresponding market segments. Table 4.14 presents an analysis of the target groups of potential customers.

Table 4.14 – Selection of Target Groups of Potential Consumers

No.	Description of Target Group Profile	Consumer Readiness to Adopt the Product	Estimated Demand Within the Target Group (Segment)	Competition Intensity in the Segment	Entry Difficulty into the Segment
1.	Individual Users	High	Medium	Low	Low
2.	Small Companies	High	High	Medium	Medium
3.	Large Companies	High	High	High	High

The analysis of target groups of potential consumers shows that all segments demonstrate high readiness to adopt passwordless authentication.

However, the intensity of competition and the difficulty of market entry increase from individual users to large corporations. This necessitates the adaptation of marketing strategies and resources for effective penetration into each segment. Specifically, to attract large companies, it is essential to emphasize the benefits of passwordless authentication, such as enhanced security and reduced password management costs. At the same time, for individual users, the focus should be on the product's convenience and ease of use.

To determine the basic development strategy for the passwordless authentication project, an analysis of the market position, target audience, and competitive characteristics was conducted. The results of this analysis are presented in Table 4.15.

Table 4.15 – Determination of the Basic Development Strategy

No.	Is the project a "first mover" in the market?	Will the company seek new consumers or attract existing ones from competitors?	Will the company copy the main characteristics of a competitor's product, and which ones specifically?	Competitive Behavior Strategy
1	No	Attracting new consumers and transitioning existing ones from competitors.	Adopting the best practices of competitors, particularly regarding reliability and accuracy.	Offensive

To effectively position our software product in the market, it is necessary to consider the requirements of the target audience and determine key competitive advantages. The results of this analysis are presented in Table 4.16.

Table 4.16 – Determination of the Positioning Strategy

No.	Target Audience Product Requirements	Basic Development Strategy	Key Competitive Positions of Own Startup Project	Selected Associations to Form the Project's Complex Position (Three Key One
1	Reliability, accuracy, ease of use	Cost optimization	Quality and efficiency	Reliability, accuracy, ease of use

In this section, the positioning strategy for the software product has been defined, which is based on three key aspects: algorithm efficiency, accuracy, and ease of use. These characteristics align with the core requirements of the target audience and highlight the competitive advantages of the startup project.

4.5 Development of the Marketing Program

To develop an effective marketing program, it is essential to clearly define the concept of the product being offered to consumers. This includes analyzing the needs of the target audience, the benefits the product provides, and its competitive advantages. Table 4.17 summarizes the previously conducted analysis of the product's competitiveness.

Table 4.17 – Definition of Key Advantages of the Potential Product Concept

No.	Need	Benefit Offered by the Product	Key Advantages Over Competitors (Existing or to be Created)
1	High-quality protection against threats	Reliable passwordless authentication that reduces the risk of account compromise	Use of advanced encryption algorithms and multi-factor authentication to ensure a high level of security
2	Universality	Compatibility with various platforms and devices, ensuring convenience of use	Flexible integration with existing security systems and the ability to customize to specific client needs

3	Support	Regular updates and prompt technical support for users	Availability of 24/7 support service and rapid response to identified vulnerabilities or client requests

Table 4.18 provides a description of the three-level product model for the startup project.

Table 4.18 – Description of the Three-Level Product Model for the Startup Project

<table>
<tr><td>Product Level</td><td colspan="2">Essence and Components</td></tr>
<tr><td>Core Product (Product by Conception)</td><td colspan="2">A passwordless authentication system that provides secure and convenient access for users to online services without the use of traditional passwords</td></tr>
<tr><td rowspan="4">II. Actual Product (Product in Real Execution)</td><td>Attributes/Characteristics:</td><td>Size:</td></tr>
<tr><td>Client Application</td><td>20 МБ</td></tr>
<tr><td>Server Module</td><td>50 МБ</td></tr>
<tr><td colspan="2">Quality: High system reliability and performance, support for multi-factor authentication, compatibility with various platforms and devices</td></tr>
<tr><td rowspan="2"></td><td colspan="2">Packaging: Electronic license with a unique activation key, documentation, and setup/usage instructions</td></tr>
<tr><td colspan="2">Brand: The developer company's name and product name, emphasizing the solution's innovativeness and security</td></tr>
</table>

III. Augmented Product (Product with Reinforcement)	Pre-sale: Demo version with limited functionality for client familiarization; consultations on system integration and setup
	Copy Protection: License binding to a specific domain or IP address; product activation via an online service with license key authenticity verification; use of encryption for code protection

Let's examine the price ranges defined using the expert method in Table 4.19.

Table 4.19 – Formation of Price Ranges

No.	Price Level of Substitute Goods (per 1000 users/month)	Price Level of Analogous Goods (per 1000 users/month)	Income Level of the Target Consumer Group	Upper and Lower Bounds for Setting the Price of the Good/Service (per 1000 users/month)
1	3000$	6000$	100000$	2000$ - 7000$

Table 4.20 illustrates the formation of the distribution system for the product.

Table 4.20 – Formation of the Distribution System

No.	Specifics of Target Clients' Purchasing Behavior	Distribution Functions to be Performed by the Product Supplier	Depth of the Distribution Channel	Optimal Distribution System
1	Purchase of a product license or signing a service provision agreement without transfer of software ownership rights	- Providing API access via a web service; - Distributing keys via web portals	Single-level distribution channel (direct contact with the end consumer)	Single-level distribution channel

Table 4.21 presents the marketing communications concept for the product.

Table 4.21 – Marketing Communications Concept

No.	Specifics of Target Client Behavior	Communication Channels Used by Target Clients	Key Positions Chosen for Positioning	Objective of the Advertising Message	Concept of the Advertising Appeal
1	Clients seek new methods to enhance security and authentication convenience	Professional conferences, specialized webinars, industry publications, social networks (LinkedIn)	Reliability, ease of implementation, compatibility with existing systems	To emphasize the advantages of passwordless authentication, particularly increased security and user convenience	"Forget passwords – adopt modern solutions for secure and simple authenticati on."

Table 4.21 reflects the marketing communications concept for the passwordless authentication project, focusing on the behavior of target clients seeking new methods to enhance security and authentication convenience. The primary communication channels are professional conferences, specialized webinars, industry publications, and social networks such as LinkedIn. The product positioning is based on reliability, ease of implementation, and compatibility with existing systems. The objective of the advertising message is to highlight the advantages of passwordless authentication, particularly increased security and user convenience.

This section presents a comprehensive marketing analysis, which is a key stage in the development of the startup project. The core concept of the project has been defined, a technology audit has been conducted to assess the feasibility of using

various technologies, and a detailed market analysis has been performed, including an evaluation of the competitive environment and identification of market trends. Additionally, risks and opportunities have been analyzed, the project's strengths and weaknesses have been identified, and target audience analysis has been conducted.

Based on the collected data, a market entry strategy and marketing program have been developed, aimed at effectively positioning the project, engaging the target audience, and achieving the set business goals. The analysis results indicate high competitiveness and economic feasibility of the project, confirming its readiness for market implementation.

5 OCCUPATIONAL SAFETY AND HEALTH

Occupational safety and health issues are considered for the design and development phase of climate data analysis and visualization system.

Occupational safety is a system of legal, socio-economic, organizational and technical, sanitary and hygienic and treatment and prevention measures and tools aimed at preserving human life, health and ability to work. Working conditions at the workplace, safety of technological processes, machines, mechanisms, equipment and other means of production, condition of collective and individual protection means used by the employee, as well as sanitary and living conditions must meet the requirements of the law. An employee has the right to refuse the assigned work if a work situation has arisen that is dangerous to his life or health or to the people around him, or to the work environment or the environment. He must immediately notify his immediate supervisor or employer. The existence of such a situation is confirmed, if necessary, by labor protection specialists of the enterprise with the participation of a representative of the trade union of which he is a member or a person authorized by employees on labor protection (if the trade union was not established), as well as an insurance expert [12]. The task of labor protection is to minimize injuries and illnesses of the employee while ensuring comfort with maximum productivity. The main objectives of labor protection are the formation of specialists with the necessary knowledge and practical skills on legal and organizational issues of labor protection, industrial sanitation, safety, fire safety.

5.1 General characteristics of the room and workplace

The development of the analysis and visualization system is performed in a room located on the fourth floor of an eight-storey building with general and local lighting. The room has one-sided lighting, the windows are oriented to the east, the windows have shutters. White ceiling with a reflection coefficient of 0.7, light brick walls with a reflection coefficient of 0.5. There are 4 people working in the room, in accordance with this we obtain input data for the analysis of potentially dangerous and harmful production factors, which are given in table. 4.1.

Table 5.1 - Incoming data

Room parameters	**Value**
Length x width x height	6.6 x 6.1 x 2.7 m
Area	40.26m²
Volume	108,70 m³
Workplace number	**Specifics of work**
I workplace	Front-end programmer (web application client development specialist)
II workplace	Back-end programmer (specialist in the development of the server part of web applications and database design)
III workplace	Business analyst (also acts as a product manager)
IV workplace	UI-UX web designer
Technical means (quantity)	**Name and characteristics**
Monitor (4 pcs.)	HP 22Xi / 21.5 "/ 1920x1080px / IPS
Computer (4 pcs.)	HP ProBook 440 G6, 14 "IPS screen (1920x1080) Full HD, Intel Core i7-8565U (1.8 - 4.6 GHz) / RAM 16 GB / SSD 256 GB
Floor cooler (1 piece)	CRYSTAL YLR3-5V208
Air conditioner (1 piece)	DEKKER DSH105R / G / 26m2 / 2,65kW- 2.9 kW / 25x74.5x19.5 cm / 9 kg
General purpose luminaries (3 pcs.)	The lamp raster built-in 4x18W
Local lamps (4 pcs.)	Delux Decor TF-05/1 x 40W

According to NPAOP 0.00-7.15-18, the area S 'allocated for one workplace with a personal computer must be at least 6 m2 and the volume - at least 20 m3. There are 4 workplaces in the room, which fully meets the required standards.

We calculate the actual values of these indicators by dividing the volume of the

room and the total area by the number of employees.

Therefore, based on the results obtained in terms of area and volume, the room meets the standards.

Table 5.2 - Workplace characteristics

№	The name of the parameter	Value	
		in fact	Normative
1.	Height of a working surface, mm	780	680 – 800
2.	Width of a working surface, mm	1500	not less than 600
3.	Depth of a working surface, mm	750	not less than 600
4.	Height of space for legs, mm	750	not less than 600
5.	Width of space for legs, mm	800	not less than 500
6.	Depth of space for legs, mm	750	not less than 450
7.	Seat surface height, mm	480	400 – 500
8.	Seat width, mm	500	not less than 400
9.	Seat depth, mm	500	not less than 400
10.	Height of a basic surface of a back, mm	550	not less than 300
11.	Width of a surface of a back, mm	470	Not less than 380
12.	Length of armrests, mm	300	not less than 250
13.	Width of armrests, mm	60	50 – 70
14.	Distance from eyes to the screen, mm	650	600 – 700

It is possible to draw a conclusion that the sizes of a workplace of the programmer correspond to the established norms, proceeding from the set parameters.

5.2 Analysis of potentially dangerous and harmful production factors in the workplace

When creating a system of analysis and visualization, the work is performed sitting without physical effort, so it belongs to the category of light Ia.

Premises for work must be equipped with heating, air conditioning or supply and exhaust ventilation in accordance with DBN B.2.5-67: 2013. Normalized parameters of the microclimate, ionic composition of air, content of harmful substances meet the requirements of LTO 3.3.6.042-99, GN 2152-80, GOST 12.1.005-88, DSTU GOST 12.0.230: 2008 and DSTU GOST 12.4.041: 2006. Ventilation is understood as a set of measures and means designed to ensure meteorological conditions and cleanliness of the air environment that meet hygienic and technical requirements at permanent places and service areas. The main task of ventilation is to remove polluted, humid or heated air from the room and supply clean fresh air.

The sources of noise in the room are the fan of the system unit, laptop and air conditioner. The sound generated by the fan and air conditioner can be classified as constant.

According to DBN B.2.5-28: 2018 the work belongs to the category of visual works. The use of natural, artificial and mixed lighting is envisaged.

The computer is a single-phase consumer of electricity powered by 220V AC from a network with grounded neutral. IBM PC refers to electrical installations up to 1000V closed version; all conductive parts are in the casings. According to the method of protecting a person from electric shock, computers and peripherals must meet 1 class of protection.

Technical methods of protection against electric shock is reduced to the use of current of safe voltage, protection in case of accidental touching current-carrying parts and against excessive currents, protection in case of voltage transfer to non-

current-carrying metal parts of the installation.

Safe voltage is obtained from the high voltage grid (110-120 V) by means of step-down transformers.

Protection against contact with live parts of the installation is achieved by means of insulation, fencing off the use of blocking safety devices and inaccessibility of the location of the installations.

Switchboards are placed in closed metal casings-boxes.

Safety alarm is used in the form of posters and inscriptions. The best light alarms are double, which in the presence of voltage lights a red light, and in its absence - green.

CONCLUSIONS

This work developed and tested a passwordless authentication system based on modern WebAuthn and FIDO2 standards. The system ensures resilience to phishing and man-in-the-middle (MITM) attacks and enhances user authentication convenience through the integration of asymmetric cryptography and the use of hardware tokens [40].

For the WebAuthn integration, the Yubico library was chosen, which streamlined the development process and ensured compliance with security standards. A model of the passwordless authentication system was created, encompassing user registration mechanisms, authentication using public and private keys, and cryptographic key management.

A test system was deployed, with configured processes for registration, login, and access recovery. Additionally, configuration for secure SSL connection, integration with a PostgreSQL database, and support for flexible setup via environment variables were implemented.

The proposed solution significantly enhances user security and reduces risks associated with password usage, as passwords are entirely eliminated from the authentication process. Furthermore, the integration of biometric data and hardware tokens provides an additional layer of protection and makes the system more convenient for end users.

REFERENCES

1. America's Password Habits | Security.org. Security.org. URL: https://www.security.org/resources/online-password-strategies/ (accessed: 03.10.2024).

2. Das S., Phelan L. A., Hoyos-Rivera J. A. Password Managers Usage and Trust: A US Study of User Behavior, Preferences, and Perceptions. ACM Transactions on Privacy and Security. Vol. 24, no. 3, article 16, July 2021. URL: https://doi.org/10.1145/3447564 (date of access: 09.10.2024).

3. Saul Johnson, Jo˜ao F. Ferreira, Alexandra Mendes, and Julien Cordry. “Skeptic: Automatic, justified and privacy-preserving password composition policy selection”. In: Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS '20), October 2020.

4. What is WebAuthn? Authentication standard. Wallarm | Integrated App and API Security Platform. URL: https://www.wallarm.com/what/webauthn-web-authentication (accessed: 05.10.2024).

5. Gordin, A. Graur, S. Vlad and C. I. Adomniței, "Moving forward passwordless authentication: challenges and implementations for the private cloud," 20th RoEduNet Conference: Networking in Education and Research (RoEduNet), 2021, pp. 1-5, doi: 10.1109/RoEduNet54112.2021.9638271 (date of access: 09.10.2024).

6. Decoding How WebAuthn Works. FusionAuth. URL: https://fusionauth.io/articles/authentication/webauthn (accessed: 05.10.2024).

7. What is the Passkey and how it works? Simple explanation | Hideez. Passwordless Workforce Identity Solutions | Hideez. URL: https://hideez.com/uk-ua/blogs/news/what-is-a-passkey (accessed: 25.11.2024).

8. Rolfe B. Synced vs Device-Bound Passkeys: How User Convenience and Authentication Experiences Vary. Authsignal - Drop-in Passkeys & Passwordless Authentication. URL: https://www.authsignal.com/blog/articles/synced-vs-device-bound-passkeys-convenience-and-authentication-experiences (date of access: 25.11.2024).

9. Device-Bound vs. Synced Passkeys (SCA & Passkeys I). Corbado - Add passkeys to any new or existing app. URL: https://corbado.com/blog/device-bound-synced-passkeys (date of access: 25.11.2024).

10. HYPR. What is a FIDO Platform Authenticator? | Security Encyclopedia. HYPR: Identity Security & Passwordless Authentication Solution. URL: https://www.hypr.com/security-encyclopedia/platform-authenticator (date of access: 25.11.2024).

11. Platform vs Cross-Platform. Yubico Developers. URL: https://developers.yubico.com/WebAuthn/WebAuthn_Developer_Guide/Platform_vs_Cross-Platform.html (date of access: 25.11.2024).

12. NIST SP 800-53 Rev. 5: Security and Privacy Controls for Information Systems and Organizations. Independently Published, 2022, 462 p.

13. Blokdyk G. Passwordless Authentication, Second Edition. 5STARCooks, 2021. pp. 80-93.

14. Passwordless Authentication Methods - Wise IT Ukraine. Wise IT Ukraine. URL: https://wiseit.com.ua/metody-autentyfikacziyi-bez-parolya-vid-fudo/ (accessed: 30.09.2024).

15. A Study on Passwordless Authentication Technology and Its Effects. The International Journal of Reliable Information and Assurance. 2018. Vol. 6, no. 1. URL: https://doi.org/10.21742/ijria.2018.6.1.03 (date of access: 19.11.2024).

16. Passwordless Magic Links vs. Certificates for Secure Access. SecureW2. URL: https://www.securew2.com/blog/passwordless-magic-link-authentication-explained (date of access: 25.11.2024).

17. Java: Basic Principles and Benefits of Use. www.mathros.net.ua - Website for Computer Science Students. URL: https://www.mathros.net.ua/basic-principles-of-java-programming.html (accessed: 25.11.2024).

18. Java: The Complete Reference, Thirteenth Edition. McGraw-Hill Education, 2023.

19. Horstmann C. S. Core Java, Volume II--Advanced Features (11th Edition). Prentice Hall, 2019. pp. 26-28.

20. Scarioni C., Nardone M. Pro Spring Security: Securing Spring Framework 5 and Boot 2-based Java Applications. Apress, 2019. 428 p.

21. Spring Boot. Spring Boot. URL: https://spring.io/projects/spring-boot (date of access: 25.11.2024).

22. Spring Data. Spring Data. URL: https://spring.io/projects/spring-data (date of access: 25.11.2024).

23. Macero García M., Telang T. Learn Microservices with Spring Boot 3. 3rd ed. Berkeley, CA : Apress, 2023. pp. 11-18.

24. Yubico Developers. Yubico Developers. URL: https://developers.yubico.com/ (date of access: 25.11.2024).

25. Tudose C. Java Persistence with Spring Data and Hibernate. Manning, 2022. 625 p.

26. Hibernate. Everything data. Hibernate. URL: https://hibernate.org/ (date of access: 25.11.2024).

27. Apache Maven Series | Baeldung. Baeldung. URL: https://www.baeldung.com/maven-series (accessed: 11.10.2024).

28. Gaba I. What is Maven: Here's What You Need to Know [Updated]. Simplilearn.com. URL: https://www.simplilearn.com/tutorials/maven-tutorial/what-is-maven (date of access: 25.11.2024).

29. What Is PostgreSQL?. Kinsta®. URL: https://kinsta.com/knowledgebase/what-is-postgresql/#what-is-postgresql (date of access: 29.09.2024).

30. Schönig H.-J. Mastering PostgreSQL 12: Advanced Techniques to Build and Administer Scalable and Reliable PostgreSQL Database Applications, 3rd Edition. Packt Publishing, Limited, 2019.

31. Mastering PostgreSQL 15: Advanced techniques to build and manage scalable, reliable, and fault-tolerant database applications, 5th Edition. Packt Publishing, 2023. 522 p.

32. Blog H. H. T. Passwordless Authentication With Passkey: How It Works and Why It Matters – Part 1. Medium. URL: https://medium.com/@heritage.tech/passwordless-authentication-with-passkey-how-it-works-and-why-it-matters-part-1-dcae2a004988 (date of access: 29.09.2024).

33. Web client definition Glossary. NordVPN. URL: https://nordvpn.com/uk/cybersecurity/glossary/web-client/ (date of access: 09.10.2024).

34. HYPR. What is a FIDO Relying Party (RP)? | Security Encyclopedia. Identity Security & Passwordless Authentication Solution | HYPR. URL: https://www.hypr.com/security-encyclopedia/relying-party-rp (date of access: 05.10.2024).

35. Is FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication / S. Ghorbani Lyastani et al. 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2020. 2020. URL: https://doi.org/10.1109/sp40000.2020.00047 (date of access: 19.11.2024).

36. Biometrics, FIDO, and More: A Guide to Passwordless Authentication Methods. BIO-key Blog. URL: https://blog.bio-key.com/biometrics-fido-and-more-a-guide-to-passwordless-authentication-methods (accessed: 09.10.2024).

37. What is a relational database?. Oracle | Cloud Applications and Cloud Platform. URL: https://www.oracle.com/database/what-is-a-relational-database/ (date of access: 09.10.2024).

38. Does my browser support WebAuthn?. Does my browser support WebAuthn?. URL: https://webauthn.me/browser-support (date of access: 25.11.2024).

39. Rasmussen B. A Usability Study of FIDO2 Roaming Software Tokens as a Password Replacement. 2021. URL: https://scholarsarchive.byu.edu/etd/9227 (date of access: 19.11.2024).

40. What is Passwordless Authentication and How Does It Work?. LoginTC. URL: https://www.logintc.com/types-of-authentication/passwordless-authentication/ (date of access: 09.10.2024).

Appendix A


EUROPEAN CONFERENCE
Conference Proceedings
III International Science Conference
«Technology development: shaping modern thinking and scientific approaches»
January 19-21, 2026
Krakow, Poland

TABLE OF CONTENTS

МОДЕЛЮВАННЯ ТА АНАЛІЗ БЕЗПАРОЛЬНОЇ АВТЕНТИФІКАЦІЇ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ТА ВИЯВЛЕННЯ АНОМАЛІЙ У ПОВЕДІНЦІ КОРИСТУВАЧІВ

Голотенко Олександр Сергійович
кандидат технічних наук, доцент
Тернопільський національний технічний університет імені Івана Пулюя

Акінємі Віктор Олувасеї
здобувач другого (магістерського) рівня вищої освіти, 6 курс
Тернопільський національний технічний університет імені Івана Пулюя

Касонго Валері Бванга
здобувач другого (магістерського) рівня вищої освіти, 6 курс
Тернопільський національний технічний університет імені Івана Пулюя

Паролі традиційно слугують основним засобом автентифікації, проте вони стали «слабкою ланкою» кібербезпеки. За даними звіту Verizon, до 81% випадків зламів пов'язані з використанням слабких або викрадених паролів [1]. Це стимулює перехід до безпарольної автентифікації, яка усуває залежність від статичних секретів (паролів) та мінімізує ризики, пов'язані з людським фактором. Безпарольні методи (біометричні дані, криптографічні ключі, одноразові коди тощо) пропонують підвищену безпеку і зручність для користувачів, усуваючи необхідність спільних секретів при вході в систему [2]. Однак впровадження таких методів у масштабі організації створює нові виклики – наприклад, як впевнитись, що автентифікований без пароля користувач дійсно є тим, за кого себе видає, протягом усього сеансу. Тут на допомогу приходять методи штучного інтелекту та машинного навчання. Застосування аномалійного моніторингу поведінки користувачів дає змогу постійно валідувати особу користувача після початкової автентифікації, гарантуючи, що доступ підтримується лише для легітимного користувача [3]. Таким чином, поєднання безпарольної автентифікації з алгоритмами машинного навчання для виявлення аномалій здатне суттєво підвищити рівень безпеки систем автентифікації.

Безпарольна автентифікація: концепції та стандарти

Безпарольна автентифікація – це підхід, за якого користувач отримує доступ до системи без введення звичного пароля. Натомість використовуються інші фактори автентифікації, наприклад біометрія (відбиток пальця, розпізнавання обличчя), апаратні токени або одноразові коди. Одним із найбільш відомих сучасних стандартів є FIDO2 (Fast Identity Online 2), розроблений альянсом FIDO

спільно з консорціумом W3C. Стандарт FIDO2 базується на криптографії з відкритим ключем: під час реєстрації генерується пара ключів (приватний зберігається на пристрої користувача, публічний – на сервері), і надалі для входу сервер надсилає випадковий «виклик», який підписується приватним ключем на пристрої користувача [4]. Замість пароля користувач підтверджує свою особу тим самим способом, що й розблоковує свій пристрій – наприклад, відбитком пальця, сканом обличчя або PIN-кодом [4]. Такий підхід усуває ризики фішингу, повторного використання та перехоплення паролів, оскільки жодні секрети не передаються і не зберігаються на сервері [5]. В результаті безпарольна автентифікація забезпечує вищий рівень захисту облікових записів і покращує користувацький досвід (немає потреби запам'ятовувати чи регулярно змінювати паролі).

Попри переваги, безпарольна автентифікація не усуває всіх можливих загроз. Залишається ризик скомпрометувати самі пристрої або токени користувачів, викрасти їх або обійти біометрію. Також зловмисник може спробувати отримати доступ, навіть якщо автентифікація пройдена (наприклад, використовуючи автоматизовані сесії або атакуючи вже залогінених користувачів). Тому виникає потреба у додатковому рівні моніторингу після автентифікації – зокрема, шляхом аналізу поведінкових факторів користувача. Саме тут доречним є застосування методів машинного навчання для виявлення аномалій у поведінці в режимі реального часу.

Машинне навчання для виявлення аномалій поведінки користувачів

Аномалія поведінки – це відхилення у діях користувача від притаманного йому звичного патерну. Системи виявлення аномалій на основі машинного навчання навчаються розуміти, що є «нормальною» поведінкою для конкретного користувача або групи користувачів, і сигналізують при виявленні суттєвого відхилення від цієї норми. Такий підхід належить до концепції UEBA (User and Entity Behavior Analytics) – аналітики поведінки користувачів та сутностей, яка поєднує статистичні методи і алгоритми ML для фіксації нетипових дій.

Для виявлення аномалій використовують різні підходи машинного навчання залежно від наявності даних та типових сценаріїв атак. Коротко розглянемо основні методи та алгоритми:

- Наглядове навчання (supervised): модель тренується на заздалегідь розмічених даних, де відомо, які дії були нормальними, а які – зловмисними.
- Безнаглядове навчання (unsupervised): алгоритми самостійно виявляють структуру в даних, групуючи схожі сеанси та вирізняючи ті, що не належать до жодної групи.
- Напівнаглядове навчання: комбінує два підходи – модель вчиться переважно на нормальних (немічених) даних, маючи лише невелику кількість відомих аномальних прикладів.
- Глибоке навчання: нейронні мережі, зокрема рекурентні (RNN, LSTM), використовуються для аналізу послідовностей дій користувача у часі.

Конкретні алгоритми, що зарекомендували себе для задач виявлення аномалій у користувацькій активності, включають Isolation Forest, One-Class

SVM, Autoencoder (автоенкодери) та методи кластеризації на кшталт DBSCAN і k-means.

Поведінкові характеристики, що аналізуються, залежать від доступних даних системи автентифікації. Зазвичай враховуються такі фактори, як: географічне розташування входу, час доби та день тижня активності, частота та тривалість сеансів, тип і стан пристрою, з якого здійснено вхід, перелік дій або ресурсів, до яких звертається користувач, швидкість і ритм введення даних тощо. На основі цих даних будується профіль нормальної поведінки. Як зазначають дослідники, ефективним є підхід побудови "відбитку сесії" (session fingerprint) – агрегованого профілю, що характеризує поведінку користувача протягом сеансу або серії сеансів. Надалі аномалії визначаються як відхилення від цього профілю: наприклад, якщо користувач зазвичай працює у офісі з корпоративного ноутбука вдень, то спроба доступу вночі з іншої країни з незнайомого пристрою буде значущою аномалією. Система UEBA, впроваджена на рівні служби ідентифікації, може аналізувати такі "відбитки" у реальному часі та надсилати тривожні сповіщення при виявленні нетипової поведінки.

Моделювання інтегрованої системи автентифікації

На основі розглянутих технологій можна запропонувати теоретичну модель інтегрованої системи, що поєднує безпарольну автентифікацію та аномалійну аналітику поведінки. Ця система працюватиме у кілька етапів:

1. Початкова автентифікація без пароля. Користувач проходить автентифікацію за допомогою вибраного безпарольного механізму – наприклад, використовуючи апаратний ключ FIDO2 або біометричний фактор через протокол WebAuthn. На цьому етапі перевіряється криптографічний підпис виклику, що гарантує справжність фактору (ключа або біометрії) користувача [4]. Якщо перевірка пройдена, користувач отримує доступ до системи (починається сесія).

2. Моніторинг поведінки та збір даних. Після входу система починає збір телеметрії про дії користувача у сесії. Збираються такі дані, як час та тривалість активності, IP-адреса та геолокація, інформація про пристрій та браузер, список ресурсів або функцій, до яких звертається користувач, та інші поведінкові характеристики (наприклад, динаміка введення даних). Ці дані використовуються для формування поведінкового профілю сеансу.

3. Аналіз та прогнозування аномалій (машинне навчання). Зібрані дані надходять до модуля аналізу, де ML-модель оцінює, наскільки поточна поведінка відповідає нормальній для цього користувача (або для подібних користувачів). Модель може бути, скажімо, нейронною мережею або ансамблем алгоритмів (Isolation Forest + кластеризація), натренованих на попередніх сесіях. В реальному часі обчислюється метрика «ризику» або аномальності сесії. Аномалія прогнозується, якщо метрика виходить за поріг: це означає, що поточна поведінка статистично малоймовірна і може вказувати на загрозу (викрадений токен, дії зловмисника тощо).

4. Реакція та прийняття рішень. Якщо виявлено аномальну поведінку, система в режимі реального часу виконує наперед визначені дії. Можливі реакції:

запит повторної автентифікації, обмеження доступу до окремих чутливих ресурсів, повне завершення сеансу та сповіщення служби безпеки. В разі незначних відхилень може застосовуватися step-up authentication – користувачу надсилається запит підтвердити особу додатково, тоді як при критичних аномаліях сесію негайно блокують. Паралельно інцидент логуються для подальшого аналізу. Якщо ж поведінка в межах норми, користувач продовжує роботу без перешкод, і система невпинно навчається – оновлює профіль новими даними, підлаштовуючи модель під еволюцію поведінки.

Такий підхід реалізує концепцію «Zero Trust» – ніколи не довіряти повністю, навіть після успішного входу. Безперервна автентифікація на основі поведінкових ознак дозволяє ловити атакуючих вже всередині системи, коли вони імітують легітимних користувачів.

Важливо зазначити, що при моделюванні такої системи слід врахувати баланс між чутливістю виявлення та кількістю хибних спрацювань. Надто сувора модель може позначати аномалію там, де її насправді немає (наприклад, відрядження користувача в іншу країну), що призведе до зайвих блокувань і скарг. Натомість надто толерантна модель може пропустити реальну атаку. Для вирішення цієї проблеми використовують кілька підходів. По-перше, моделі регулярно перенавчають на актуальних даних, щоб вони враховували зміни у поведінці користувачів і нові типи атак. По-друге, впроваджують багаторівневий аналіз: автоматичний алгоритм виявляє сирі аномалії, але остаточне рішення приймається з урахуванням додаткового контексту або експертної оцінки. Зокрема, у новітніх наукових підходах пропонується після автоматичного класифікатора (наприклад, Isolation Forest + DBSCAN для групування аномальних сесій) застосовувати аналіз спеціаліста з безпеки: експерт переглядає кластеризовані аномальні сесії і допомагає відрізнити справді небезпечні інциденти від помилкових спрацювань. Така комбінована людино-машинна модель дозволяє досягти високої точності

Висновки

Перехід до безпарольної автентифікації є сучасною відповіддю на проблеми, пов'язані зі слабкими паролями і людським фактором у безпеці. Стандарти на кшталт FIDO2 демонструють, що можна забезпечити зручний та надійний вхід користувачів без використання пароля, позбавивши зловмисників улюбленої цілі – крадіжки або відгадування секретної фрази. Утім, впровадження безпарольних рішень потребує доповнення механізмами інтелектуального аналізу поведінки. Методи машинного навчання, інтегровані в систему автентифікації, дозволяють прогнозувати та виявляти аномалії на основі поведінкових патернів користувачів. Це суттєво підсилює захист: навіть якщо зловмисник обійде початкову автентифікацію (або отримає доступ до токена), його нетипова діяльність буде помічена і зупинена.

Розроблена теоретична модель демонструє, як можна поєднати переваги безпарольних технологій (відсутність паролів, фішингостійкість) з потужністю AI-систем для безперервного моніторингу. Машинне навчання автоматично будує профілі нормальної поведінки і оновлює їх зі зростанням обсягів даних,

тим самим адаптуючи безпекові політики до кожного користувача. Така адаптивна система здатна виявити навіть ті загрози, що раніше не зустрічалися, забезпечуючи проактивний захист (прогнозування інцидентів до їхнього розвитку).

Список літератури

1. Dev Kumar (2025, June 20). Machine Learning for Anomaly Detection in IAM, Passwordless, Threat, and Breach Scenarios. MojoAuth. URL: https://mojoauth.com/ciam-101/machine-learning-anomaly-detection-iam-passwordless-security

2. Portnox (2025, June 5). Passwordless Authentication and AI: A Look at Emerging Technologies. Portnox Blog. URL: https://www.portnox.com/blog/network-security/passwordless-authentication-and-ai-a-look-at-emerging-technologies/

3. Martín, A. G., Beltrán, M., Fernández-Isabel, A., & Martín de Diego, I. (2021). An approach to detect user behaviour anomalies within identity federations. Computers & Security, 108, 102356. https://doi.org/10.1016/j.cose.2021.102356

4. Lindemulder, G., & Kosinski, M. (n.d.). What is FIDO2? IBM Think Blog. URL: https://www.ibm.com/think/topics/fido2

5. Perapu, P. (2025). Anomaly Detection in User Behaviour Using Machine Learning for Cloud Platforms. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 11(3), 805-809.