

VISTULA UNIVERSITY

The Faculty of Computer Engineering, Graphic Design and Architecture

Program of study Computer Science

Jean-Hervé Ingabire

Student's number 65741

***DEVELOPING WEB AND MOBILE
APPLICATIONS FOR PRODUCT
AUTHENTICATION***

The master's thesis
written under the supervision of
Dr. Selcuk Cankurt

Warsaw, 2024

ABSTRACT

Fake products are a major problem in many industries, such as fashion, electronics, and pharmaceuticals. They are affecting both consumers, brands, market and businesses in general. They can cause health risks, economic losses, and damage to brand reputation. They not only cheat consumers of their hard-earned money, but also pose serious risks to their health and safety.

Imagine buying a fake medicine that does nothing to cure your illness, or worse, makes it worse. Imagine buying a fake car part that fails at a crucial moment, causing a fatal accident. Imagine buying a fake toy that contains toxic materials, harming your child. These are not hypothetical scenarios, but real cases that happen every day around the world.

How can we detect fake products and distinguish them from real ones? This is the main question that motivated my thesis subject research.

TABLE OF CONTENTS

ABSTRACT.....	1
TABLE OF CONTENTS.....	2
LIST OF FIGURES.....	5
CHAP I. INTRODUCTION 1.1 Background.....	6
1.2 Research 1.2.1 Research questions.....	6
1.2.2 Research goals.....	6
1.2.3 Organization on the Thesis.....	8
CHAP II. LITERATURE REVIEW.....	9
2.1 Web application.....	9
2.1.1 The Web in perspective.....	9
2.1.2 The origins of the Web.....	9
2.1.3 What's a Web application.....	10
2.1.4 How do web applications work?.....	10
2.1.4.2 Server-side architecture.....	11
2.1.5 Advantages and disadvantages of web applications.....	12
2.1.5.2 Disadvantages.....	12
2.2 Mobile application.....	12
2.2.1 The Evolution of Mobile Applications.....	12
2.2.2 Common Mobile Application Functions.....	13
2.2.3 Types of mobile applications.....	15
2.2.3.1 Native applications.....	15
2.2.4 Advantages and disadvantages of mobile applications.....	15
2.2.4.1 Advantages.....	16
2.2.4.2 Disadvantages.....	17
2.3 Blockchain.....	17
2.3.1 What's a Blockchain?.....	17
2.3.2 The Purpose of the Blockchain.....	18
2.3.3 The term "Blockchain".....	18
2.3.3 Types of blockchain networks.....	20
2.3.3.1 Public blockchain networks.....	20

2.3.2.2 Private blockchain networks	20
2.3.2.3 Permissioned blockchain networks	21
2.3.2.4 Consortium blockchains	21
2.3.3 Types of blockchain protocols	21
2.3.3.1 Hyperledger fabric	21
2.3.3.2 Ethereum	22
2.3.3.3 Corda.....	22
2.3.3.4 Quorum	22
2.3.4 Key elements of a blockchain	22
2.3.4.1 Distributed ledger technology	22
2.3.4.2 Immutable records	22
2.3.4.3 Smart contracts	23
2.3.5 Blockchain Architecture	23
2.3.5.1 Architecture	23
2.3.5.2 Key Characteristics of Blockchain Architecture	24
2.3.6 How does Blockchain works?	25
2.3.6.1 Proof of work vs proof of stake.....	27
2.3.7 Why blockchain is important?.....	27
2.3.8 Blockchain use cases	28
2.3.9 Is blockchain secure?.....	29
2.3.11 Disadvantages of the blockchain	31
2.4 QR Code for Product Authentication.....	31
2.4.1 What's QR Code?	31
2.4.2 Static vs dynamic QR codes.....	33
2.4.3 Usability of QR code.....	33
2.4.4 QR code components.....	33
2.4.5 Advantages and disadvantages of QR Codes	34
2.4.5.1 Advantages of QR Codes	34
2.4.5.2 Disadvantages of QR Codes	35
CHAP III. DESIGN AND IMPLEMENTATION	36
3.1 Requirements	36
3.2 High level architecture diagram.....	38

3.2.1 Web application	39
3.2.2 Mobile application	39
3.2.3 API Gateway.....	39
3.2.4 Persistent layer	40
3.2.4.1 Why combining Blockchain and MongoDB ?	41
3.3 Sequence diagram.....	42
3.4 Class Diagrams.....	43
3.5 Involved technologies and tools	44
3.5.2 Mobile application technology stacks.....	45
3.5.3 API Gateway technology stacks	46
3.6.1 Web application	49
3.6.1.1 Login.....	49
3.6.1.2 Home.....	49
3.6.1.5.1 Web users.....	54
3.6.2.3.2 Scan from Memory	62
3.6.2.5 Menu	67
CHAP IV. DISCUSSION	68
4.2 Limitations	68
4.3 Future implementations	69
CHAP V. CONCLUSION.....	71
BIBLIOGRAPHY	72
APPENDIX.....	74

LIST OF FIGURES

Figure 1: Server-side architecture	11
Figure 2: Blockchain networks	20
Figure 3: Blockchain architecture	23
Figure 4: How blockchain works	25
Figure 5: QR Code versions	32
Figure 6: High level architecture diagram	38
Figure 7: Sequence diagram	42
Figure 8: Class diagrams	43

CHAP I. INTRODUCTION

1.1 Background

First, let's see some background information on the scope and impact of fake products, objectives, methods, and contributions of my research on developing an ecosystem to detect fake products and real ones.

Fake products are products that imitate or copy the appearance, features, or quality of a genuine product, but are not authorized by the original manufacturer or seller. They can be fake, pirated, or fraudulent products.

Given the magnitude and complexity of the problem of fake products, there is a need for effective and efficient solutions to detect them and protect consumers, brands, and market. One possible solution is the one that I will be presenting further consists of developing a system that can automatically identify fake products and real ones based on combination of various modern technologies. Such a system could help consumers make informed decisions when buying a product; help brands monitor and report fake products that infringe their trademarks; help market regulators enforce intellectual property laws and combat illicit trade.

1.2 Research

1.2.1 Research questions

- How can web and mobile applications contribute to verify product authenticity?
- What are the benefits and challenges of using such a system for manufacturers and customers?
- How does such a system affect customer trust, confidence, and behavior?

1.2.2 Research goals

The aim of this thesis is to design and implement a web and mobile applications. Such a system should store data securely using Blockchain technology and MongoDB. Secure QR Code can be attached to each product so that the customer can scan it using a mobile application and get a confirmation whether the product is original or fake. Blockchain is a distributed ledger that ensures data integrity and transparency, while Secure QR Code is a cryptographic technique that embeds information into a barcode. By combining these two technologies and others that I will have the

opportunity to talk about and develop in upcoming chapters, we can create a system that can verify the authenticity of products and track their history.

The Product authentication system that I have baptized “ProAuth” is an ecosystem that plays a role of informing the customers about the authenticity of a product before purchasing it. The product should be in any kind of form either in physic or virtual one like a phone, a computer, a shoe, a book, a software, a bike, headphone, etc.

The system will consist of five main components: a web application for manufacturers/brands, a mobile application for consumers, a distributed ledger, an API gateway and a database. The smart label will be attached to the product and contain a unique identifier and a QR code. The mobile application will scan the QR code and communicate with the distributed ledger and the database to retrieve the product's information through API gateway, such as its name, reference, manufacturer, country of origin, year of release and guarantee duration. The distributed ledger will store the product's information in a secure and immutable way, ensuring its integrity and transparency. The classic database will duplicate a copy of the distributed ledger. If there is any inconsistency or anomaly, the system will alert the user that the product may be fake by scanning the QR code attached to the product.

The system will have several advantages. First, it will be more faster, reliable and accurate, as it will use multiple sources of information and sophisticated and trusted approaches to verify the product's authenticity. Second, it will be more convenient and user-friendly, as it will only require a smartphone and an internet connection to operate. Third, it will be more secure as it will store the core dataset in a distributed ledger which is immutable.

This thesis aims to propose an innovative solution for identifying and preventing the distribution of fake products in the market. The solution is based on the integration of blockchain technology, secure QR, web application and an android mobile application, which can provide a reliable and transparent way of verifying the authenticity and origin of the products.

The proposed system consists of two main components: an android app and a web application. The android app allows users to scan the QR code attached to the product and retrieve the product information from the blockchain network.

The web application allows manufacturers and distributors to register and update the product information in the blockchain network, as well as to generate and print the Secure QR Code for each product.

1.2.3 Organization on the Thesis

The thesis is organized into five chapters: Chapter 1 introduces the background and motivation of the research, Chapter 2 reviews the related work and literature, Chapter 3 describes the design and implementation of the proposed system, Chapter 4 evaluates the performance and security of the proposed system, and Chapter 5 concludes the thesis and discusses directions of the future work.

CHAP II. LITERATURE REVIEW

This chapter reviews the related literature on web and mobile applications, blockchain, QR code, and product authentication system. It discusses the advantages and challenges of using web and mobile applications, blockchain for product authentication, as well as the security and usability aspects of QR code.

2.1 Web application

2.1.1 The Web in perspective

A little more than a decade ago at CERN (the scientific research laboratory near Geneva, Switzerland), Tim Berners-Lee presented a proposal for an information management system that would enable the sharing of knowledge and resources over a computer network.

The system he proposed has propagated itself into what can truly be called a World Wide Web, as people all over the world use it for a wide variety of purposes:

- Educational institutions and research laboratories were among the very first users of the Web, employing it for sharing documents and other resources across the Internet.
- Individuals today use the Web (and the underlying Internet technologies that support it) as an instantaneous international postal service, as a worldwide community bulletin board for posting virtual photo albums, and as a venue for holding global yard sales.
- Businesses engage in e-commerce, offering individuals a medium for buying and selling goods and services over the net. They also communicate with other businesses through B2B (business-to-business) data exchanges, where companies can provide product catalogues, inventories, and sales records to other companies.

2.1.2 The origins of the Web

Tim Berners-Lee originally promoted the World Wide Web as a virtual library, a document control system for sharing information resources among researchers. Online documents could be accessed via a unique document address, a Universal Resource Locator (URL). These documents could be cross-referenced via hypertext links.

From the very beginnings of Internet technology, there has been a dream of using the Internet as a universal medium for exchanging information over computer networks. Many people shared this dream. Ted Nelson's Xanadu project aspired to make that dream a reality, but the goals were lofty and were never fully realized. Internet file sharing services (such as FTP and Gopher) and message forum services (such as Netnews) provided increasingly powerful mechanisms for this sort of information exchange, and certainly brought us closer to fulfilling those goals. However, it took Tim Berners-Lee to (in his own words) "marry together" the notion of hypertext with the power of the Internet, bringing those initial dreams to fruition in a way that the earliest developers of both hypertext and Internet technology might never have imagined. His vision was to connect literally everything together, in a uniform and universal way.

2.1.3 What's a Web application

A web application is a software program that runs on a web server and is accessed through the internet. Unlike traditional applications that run on your computer, tablet, or phone, web applications are accessed through a web browser like Microsoft Edge. They are built using web technologies such as HTML, CSS, and JavaScript. Web applications can be used for a variety of purposes, such as online shopping, social networking, and online banking.

Web applications are divided into two parts: front-end and back-end. The front-end is the part of the application that the user interacts with, while the back-end is responsible for processing data and performing tasks behind the scenes.

2.1.4 How do web applications work?

Web applications have a client-server architecture. Their code is divided into two components: client-side scripts and server-side scripts.

2.1.4.1 Client-side architecture

The client-side script deals with user interface functionality like buttons and drop-down boxes. When the end user clicks on the web app link, the web browser loads the client-side script and renders the graphic

elements and text for user interaction. For example, the user can read content, watch videos, or fill out details on a contact form. Actions like clicking the submit button go to the server as a client request.

2.1.4.2 Server-side architecture

The server-side script deals with data processing. The web application server processes the client requests and sends back a response. The requests are usually for more data or to edit or save new data. For example, if the user clicks on the Read More button, the web application server will send content back to the user. If the user clicks the Submit button, the application server will save the user data in the database. In some cases, the server completes the data request and sends the complete HTML page back to the client. This is called server-side rendering.

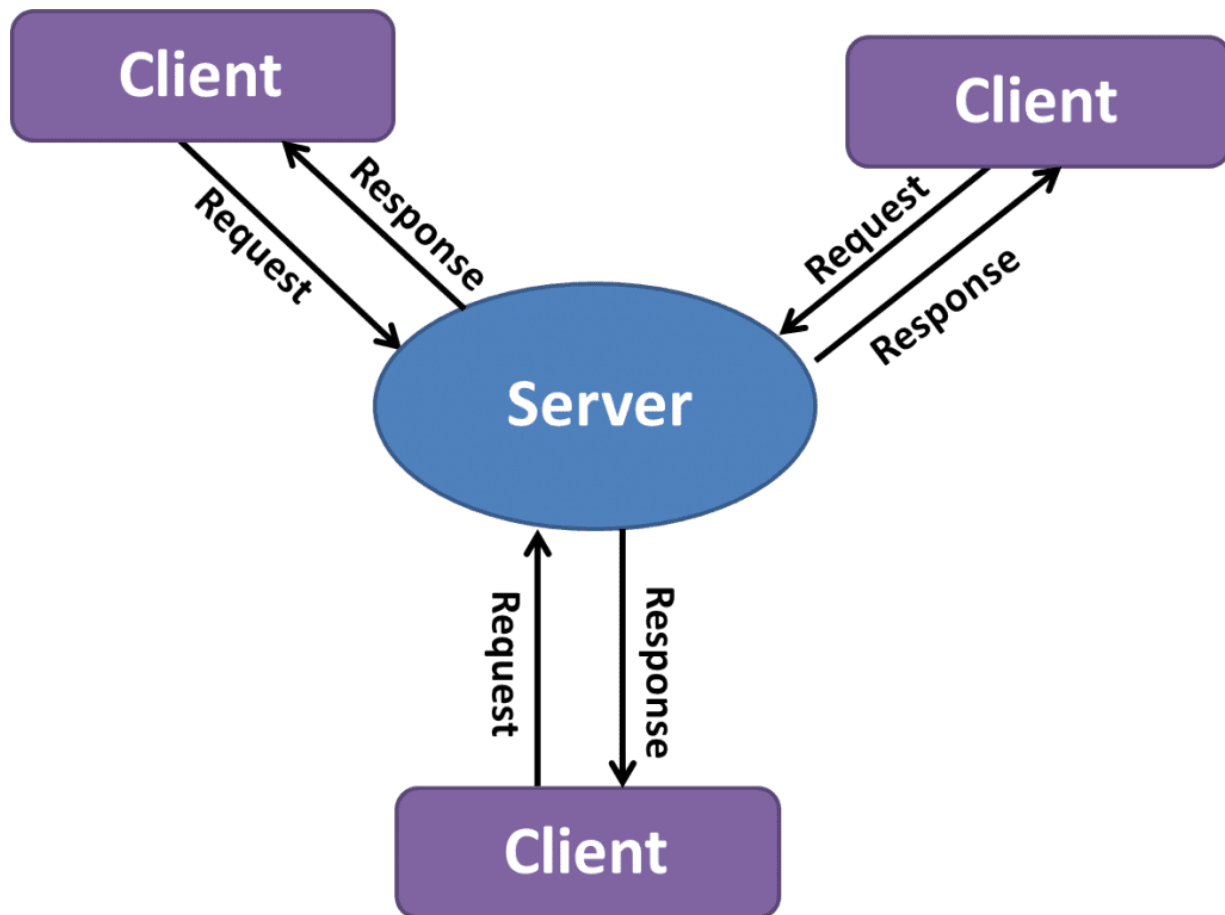


Figure 1: Server-side architecture

Source: <https://altitudeaccelerator.ca/software-architecture-design-patterns/>

2.1.5 Advantages and disadvantages of web applications

2.1.5.1 Advantages

- Easy to build
- Web apps are used less storage than other applications.
- Web Apps are preinstalled on all devices.
- Web applications are easily accessible in any type of application.

2.1.5.2 Disadvantages

- Local resources are not available in web applications.
- Depends on internet networks/connections.

2.2 Mobile application

A mobile application is a software application developed specifically for use on small, wireless computing devices, such as smartphones and tablets or watch, rather than desktop or laptop computers.

2.2.1 The Evolution of Mobile Applications

The first mobile phone applications were developed by handset manufacturers; documentation was sparse, and little information existed in the public domain on the operating internals. This can perhaps be attributed to a fear from the vendors that opening the platforms to third-party development might have exposed trade secrets in what was not yet a fully developed technology. The early applications were similar to many of the manufacturer-based apps found on today's phone, such as contacts and calendars, and simple games such as Nokia's popular Snake.

When smartphones emerged as the successor to personal digital assistants (PDAs), application development really began to take off. The growth of mobile applications can perhaps be directly attributed to the increased processing power and capabilities of the smartphone combined with the growing demand for functionality driven by the consumer market. As smartphones have evolved, mobile applications have been able to take advantage of the enhancements of the platforms. Improvements in the global positioning system (GPS), camera, battery life, displays, and processor

have all contributed to the feature-rich applications that we know today.

Third-party application development came to fruition in 2008 when Apple announced the first third-party application distribution service, the App Store. This followed on from the company's first smartphone, the iPhone, which had been released the previous year. Google closely followed with the Android Market, otherwise known today as Google Play. Today, a number of additional distribution markets exist, including the Windows Phone Store, the Amazon Appstore, and the BlackBerry World to name but a few.

The increased competition for third-party application development has left the developer markets somewhat fragmented. The majority of mobile applications are platform specific, and software vendors are forced to work with different operating systems, programming languages, and tools to provide multi-platform coverage. That is, iOS applications traditionally have been developed using Objective-C, Android, and BlackBerry applications using Java (up until BlackBerry 10, which also uses Qt) and Windows Phone applications using the .NET Framework. This fragmentation can often leave organizations requiring multiple development teams and maintaining multiple codebases.

However, a recent increase has occurred in the development of cross-platform mobile applications as organizations look to reduce development costs and overheads. Cross-platform frameworks and development of HTML5 browser-based applications have grown in popularity for these exact reasons and, in our opinion, will continue to be increasingly adopted.

2.2.2 Common Mobile Application Functions

Mobile applications have been created for practically every purpose imaginable. In the combined Apple and Google distribution stores alone, there are believed to be more than 5 million applications(in 2022) covering a wide range of functions, including some of the following:

- Online banking (Barclays)
- Shopping (Amazon)
- Social networking (Facebook)

- Streaming (Sky Go)
- Gambling (Betfair)
- Instant Messaging (WhatsApp)
- Voice chat (Skype)
- E-mail (Gmail)
- File sharing (Dropbox)
- Games (Angry Birds)

Mobile applications often overlap with the functionality provided by web applications, in many cases using the same core server-side APIs and displaying a smartphone-compatible interface at the presentation layer.

In addition to the applications that are available in the various distribution markets, mobile applications have been widely adopted in the business world to support key business functions. Many of these applications provide access to highly sensitive corporate data, including some of the following, which have been encountered by the authors during consultancy engagements:

- Document storage applications allowing users to access sensitive business documents on demand
- Travel and expenses applications allowing users to create, store, and upload expenses to internal systems
- HR applications allowing users to access the payroll, time slips, holiday information, and other sensitive functionality
- Internal service applications such as mobile applications that have been optimized to provide an internal resource such as the corporate intranet
- Internal instant messaging applications allowing users to chat in real time with other users regardless of location

In all of these examples, the applications are considered to be “internal” applications and are typically developed in-house or specifically for an organization. Therefore, many of these applications require virtual private network (VPN) or internal network access to function so that

they interact with core internal infrastructure. A growing trend in enterprise applications is the introduction of “geo fencing” whereby an application uses the device’s GPS to ascertain whether a user is in a certain location, for example, the organization’s office, and then tailors or restricts functionality based on the result.

2.2.3 Types of mobile applications

2.2.3.1 Native applications

Native applications are built for particular operating systems, which are mostly Android and IOS. Also, there are more OS for mobile applications: Blackberry and Windows. This is available for download on Google Play Store and for IOS Apple App Store. Native applications are generally built to make the most of all the features and tools of the phones such as contacts, cameras, sensors, etc. Native apps ensure high performance and stylish user experience as the developers use the native device UI to build apps.

2.2.3.2 Hybrid applications

Hybrid applications combine elements of both native and web applications, using a single codebase to run on multiple platforms. These are deployed on container that uses mobile WebView object.

2.2.3.3 Web applications

Mobile web applications are optimized for mobile devices and they are accessible through a web browser on a smartphone. These are developed using HTML, CSS and JavaScript. They runs with the help of web browser applications like chrome, safari, Firefox etc.

2.2.4 Advantages and disadvantages of mobile applications

It is not difficult to see why mobile applications have seen such an explosive rise in prominence in such a short space of time. The commercial incentives and benefits of mobile applications are obvious. They offer organizations the opportunity to reach out to end users almost all the time and to much wider audiences due to the popularity of smartphones. However, several technical factors

have also contributed to their success:

- The foundations of mobile applications are built on existing and popular protocols. In particular, the use of HTTP is widely adopted in mobile deployments and is well understood by developers.
- The technical advancements of smartphones have allowed mobile applications to offer more advanced features and a better user experience. Improvements in screen resolution and touch screen displays have been a major factor in improving the interactive user experience, particularly in gaming applications. Enhancements in battery life and processing power allow the modern smartphone to run not just one but many applications at once and for longer. This is of great convenience to end users as they have a single device that can perform many functions.
- Improvements in cellular network technologies have resulted in significant speed increases. In particular, widespread 3G and 4G coverage has allowed users to have high-speed Internet access from their smartphones. Mobile applications have taken full advantage of this to provide access to an array of online services.
- The simplicity of the core technologies and languages used in mobile development has helped with the mobile revolution. Applications can be developed using popular and mature languages such as Java, which are well understood and have a large user base.

2.2.4.1 Advantages

- Convenience: Users can access services or information on-the-go, anytime, and anywhere.
- Enhanced User Experience: Tailored interfaces and functionalities provide an engaging experience.
- Increased Engagement: Push notifications and personalized content boost user interaction.
- Offline Accessibility: Some apps allow limited functionality even without an internet connection.
- Wider Reach: Apps can attract a broader audience and create better brand visibility.

- **Monetization Opportunities:** Through ads, in-app purchases, or subscriptions, apps offer revenue channels.
- **Faster Loading:** Native apps tend to load faster and offer better performance than web-based alternatives.
- **Device Integration:** Access to device features like camera, GPS, or sensors enriches app functionality.

2.2.4.2 Disadvantages

- **Development Complexity:** Building and maintaining apps across multiple platforms can be complex and costly.
- **Platform Fragmentation:** Variations in OS versions and devices can affect app compatibility and performance.
- **App Store Approval Process:** Approval delays or rejections can hinder timely app launches or updates.
- **Security Concerns:** Mobile apps are susceptible to data breaches or unauthorized access.
- **Storage Space:** Apps can consume significant device storage, affecting user device performance.
- **User Retention:** With numerous apps available, retaining users and ensuring regular engagement can be challenging.
- **Dependency on Connectivity:** Most apps require a stable internet connection, limiting functionality in offline mode.
- **Costs:** Developing and maintaining an app can be costly, especially for advanced functionalities.

2.3 Blockchain

2.3.1 What's a Blockchain?

Blockchain is a distributed ledger technology that enables secure and transparent transactions among multiple parties without the need for a central authority or intermediary. Blockchain records transactions in blocks that are linked together by cryptographic hashes, forming a chain of immutable and verifiable records. Each block contains a timestamp, a nonce, a hash of the previous block, and a set of transactions. Transactions are validated by a consensus mechanism among the

network nodes, such as proof-of-work or proof-of-stake. Once validated, transactions are broadcasted to the network and appended to the ledger.

2.3.2 The Purpose of the Blockchain

When designing a software system, one can choose which architectural style will be used, similar to choosing an engine for a car. The architectural decision can be done independently from the functional aspects of the application layer. As a result, one can create distributed as well as centralized systems with identical functionality on the application layer. The architecture is only a means to an end when it comes to implementing a system.

Each of the two architectural concepts has its own advantages and disadvantages and their own specific way of doing things. Choosing a specific architecture has consequences on how you will achieve the functional and nonfunctional aspects of a system. In particular, both architectural concepts have very different approaches to ensure integrity. And this is the point where the blockchain enters the picture. The blockchain is a tool for achieving integrity in distributed software systems. Hence, it can be seen as a tool to achieve a nonfunctional aspect of the implementation layer.

2.3.3 The term “Blockchain”

In this discussion about the blockchain, the term is used as follows:

- As a name for a data structure
- As a name for an algorithm
- As a name for a suite of technologies
- As an umbrella term for purely distributed peer-to-peer systems with a common application area

A Data structure: In computer science and software engineering, a data structure is a way to organize data regardless of their concrete informational content. You can think about a data structure in terms of a floor plan for a building in architecture. A floor plan for a building addresses separating and connecting space with walls, floors, and stairs regardless of their concrete usage.

When used as a name for a data structure, blockchain refers to data put together into units called blocks. One can think of these blocks much like pages in a book. These blocks are connected to one another like a chain, hence the name blockchain. In relation to a book, the words and sentences are the information to be stored. They are written on different pages instead of being written on a large spool. The pages are connected with one another via their position in the book and via the page numbers. You can determine if someone removed a page from the book by checking whether the page numbers continue without leaving out a number. Furthermore, the information on the pages as well as the pages within the book are ordered. The ordering is an important detail, which will be used extensively. Additionally, the chaining of the data blocks in the data structure is achieved by using a very special numbering system, which differs from the page numbering in ordinary books.

An Algorithm: In software engineering, the term algorithm refers to a sequence of instructions to be completed by a computer. These instructions often involve data structures. When used as a name for an algorithm, blockchain refers to a sequence of instructions that negotiates the informational content of many blockchain-data-structures in a purely distributed peer-to-peer system, similar to a democratic voting schema.

A Suite of Technologies: When used to refer to a suite of technologies, blockchain refers to a combination of the blockchain-data-structure, the blockchain-algorithm, as well as cryptographic and security technologies that combined can be used to achieve integrity in purely distributed peer-to-peer systems, regardless of the application goal.

An Umbrella Term for Purely Distributed Peer-to-Peer Systems with a common Application Area: Blockchain can also be used as an umbrella term for purely distributed peer-to-peer systems of ledgers that utilize the blockchain-technology-suite. Note that in this context blockchain refers to a purely distributed system as a whole instead of referring to a software unit that is part of a purely distributed system.

2.3.3 Types of blockchain networks

There are several ways to build a blockchain network. They can be public, private and permissioned or built by a consortium.

2.3.3.1 Public blockchain networks

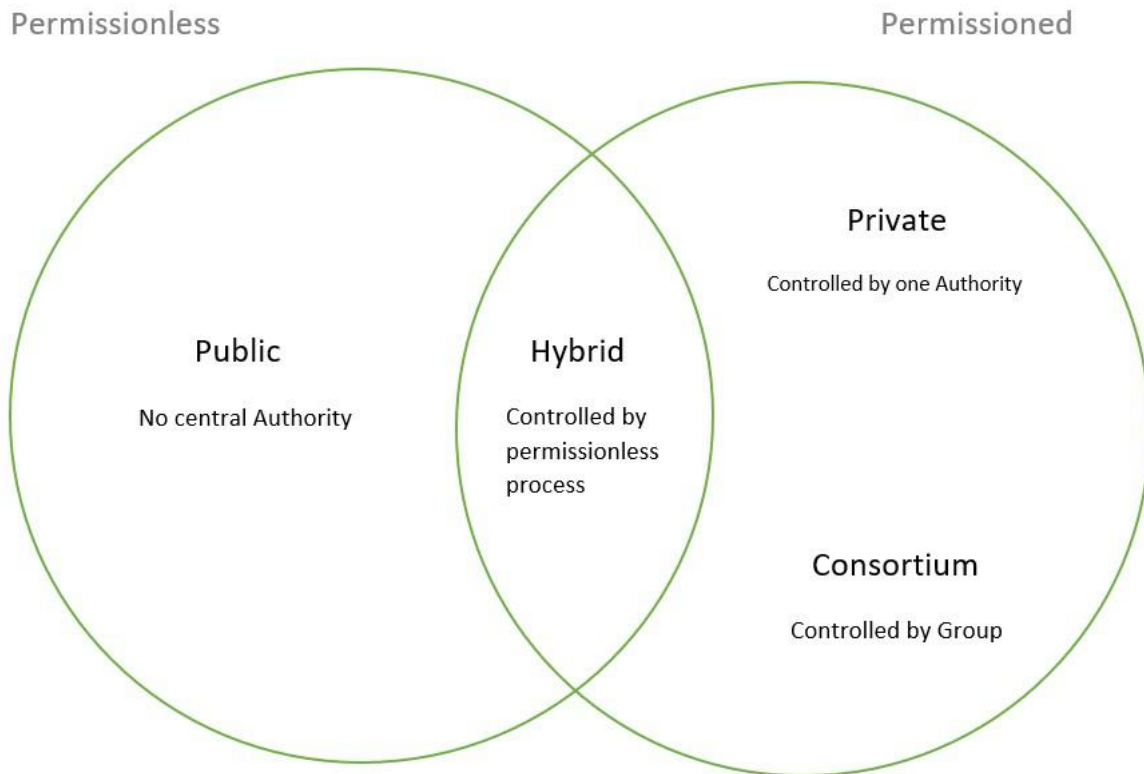


Figure 2: Blockchain networks

Source: <https://www.geeksforgeeks.org/types-of-blockchain/>

A public blockchain is one that anyone can join and participate in, such as Bitcoin. Drawbacks might include substantial computational power required, little or no privacy for transactions, and weak security. These are important considerations for enterprise use cases of blockchain.

2.3.2.2 Private Blockchain networks

A private blockchain network, similar to a public blockchain network, is a decentralized peer-to-peer network. However, one organization governs the network, controlling who is allowed to

participate, execute a consensus protocol and maintain the shared ledger. Depending on the use case, this can significantly boost trust and confidence between participants. A private blockchain can be run behind a corporate firewall and even be hosted on premises.

2.3.2.3 Permissioned blockchain networks

Businesses who set up a private blockchain will generally set up a permissioned blockchain network. It is important to note that public blockchain networks can also be permissioned. This places restrictions on who is allowed to participate in the network and in what transactions. Participants need to obtain an invitation or permission to join.

2.3.2.4 Consortium blockchains

Multiple organizations can share the responsibilities of maintaining a blockchain. These pre-selected organizations determine who may submit transactions or access the data. A consortium blockchain is ideal for business when all participants need to be permissioned and have a shared responsibility for the blockchain.

2.3.3 Types of blockchain protocols

The term blockchain protocol refers to different types of blockchain platforms that are available for application development. Each blockchain protocol adapts the basic blockchain principles to suit specific industries or applications. Some examples of blockchain protocols are provided in the following subsections:

2.3.3.1 Hyperledger fabric

Hyperledger fabric is an open-source project with a suite of tools and libraries. Enterprises can use it to build private blockchain applications quickly and effectively. It is a modular, general-purpose framework that offers unique identity management and access control features. These features make it suitable for various applications, such as track-and-trace of supply chains, trade finance, loyalty and rewards, and clearing settlement of financial assets.

2.3.3.2 Ethereum

Ethereum is a decentralized open-source blockchain platform that people can use to build public blockchain applications. Ethereum Enterprise is designed for business use cases.

2.3.3.3 Corda

Corda is an open-source blockchain project designed for business. With Corda, you can build interoperable blockchain networks that transact in strict privacy. Businesses can use Corda's smart contract technology to transact directly, with value. Most of its users are financial institutions.

2.3.3.4 Quorum

Quorum is an open-source blockchain protocol that is derived from Ethereum. It is specially designed for use in a private blockchain network, where only a single member owns all the nodes, or in a consortium blockchain network, where multiple members each own a portion of the network.

2.3.4 Key elements of a blockchain

2.3.4.1 Distributed ledger technology

All network participants have access to the distributed ledger and its immutable record of transactions. With this shared ledger, transactions are recorded only once, eliminating the duplication of effort that's typical of traditional business networks.

2.3.4.2 Immutable records

No participant can change or tamper with a transaction after it has been recorded to the shared ledger. If a transaction record includes an error, a new transaction must be added to reverse the error, and both transactions are then visible.

2.3.4.3 Smart contracts

To speed transactions, a set of rules called a smart contract is stored on the blockchain and executed automatically. A smart contract can define conditions for corporate bond transfers, include terms for travel insurance to be paid and much more.

2.3.5 Blockchain Architecture

2.3.5.1 Architecture

Blockchain is a technology where multiple parties involved in communication can perform different transactions without third-party intervention. Verification and validation of these transactions are carried out by special kinds of nodes.

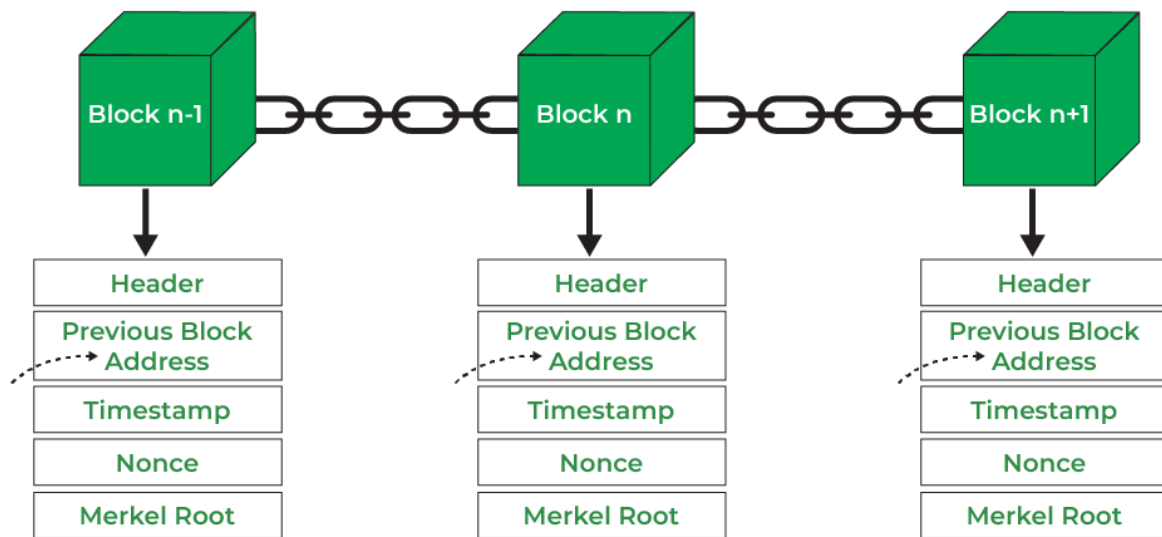


Figure 3: Blockchain architecture

Source: <https://www.geeksforgeeks.org/blockchain-structure/>

1. Header: It is used to identify the particular block in the entire blockchain. It handles all blocks in the blockchain. A block header is hashed periodically by miners by changing the nonce value as part of normal mining activity, also Three sets of block metadata are contained in the block header.

2. **Previous Block Address/ Hash:** It is used to connect the $i+1^{\text{th}}$ block to the i^{th} block using the hash. In short, it is a reference to the hash of the previous (parent) block in the chain.
3. **Timestamp:** It is a system verify the data into the block and assigns a time or date of creation for digital documents. The timestamp is a string of characters that uniquely identifies the document or event and indicates when it was created.
4. **Nonce:** A nonce number which uses only once. It is a central part of the proof of work in the block. It is compared to the live target if it is smaller or equal to the current target. People who mine, test, and eliminate many Nonce per second until they find that Valuable Nonce is valid.
5. **Merkel Root:** It is a type of data structure frame of different blocks of data. A Merkel Tree stores all the transactions in a block by producing a digital fingerprint of the entire transaction. It allows the users to verify whether a transaction can be included in a block or not.

2.3.5.2 Key Characteristics of Blockchain Architecture

- **Decentralization:** In centralized transaction systems, each transaction needs to be validated in the central trusted agency (e.g., the central bank), naturally resulting in cost and the performance jam at the central servers. In contrast to the centralized mode, a third party is not needed in the blockchain. Consensus algorithms in blockchain are used to maintain data stability in a decentralized network.
- **Persistency:** Transactions can be validated quickly and invalid transactions would not be admitted by persons or miners who mining the crypto. It is not possible to delete or roll back transactions once they are included in the blockchain network. Invalid transactions do not carry forward further.
- **Anonymity:** Each user can interact with the blockchain with a generated address, which does not disclose the real identity of the miner. Note that blockchain cannot guarantee perfect privacy preservation due to the permanent thing.
- **Auditability:** Blockchain stores data of users based on the Unspent Transaction Output (UTXO) model.

Every transaction has to refer to some previous unspent transactions. Once the current transaction is recorded into the blockchain, the position of those referred unspent

transactions switches from unspent to spent. Due to this process, the transactions can be easily tracked and not harmed between transactions.

- **Transparency:** The transparency of blockchain is like cryptocurrency, in bitcoin for tracking every transaction is done by the address. And for security, it hides the person's identity between and after the transaction. All the transactions are made by the owner of the block associated with the address, this process is transparent and there is no loss for anyone who is involved in this transaction.
- **Cryptography:** The blockchain concept is fully based on security and for that, all the blocks on the blockchain network want to be secure. And for security, it implements cryptography and secures the data using the cipher text and ciphers.

2.3.6 How does Blockchain works?

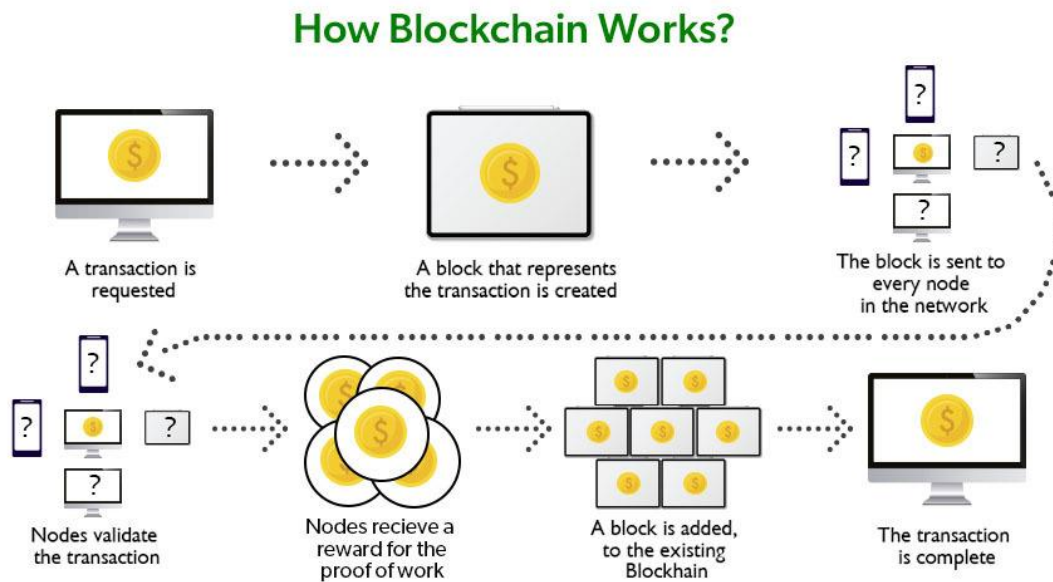


Figure 4: How blockchain works

Source: <https://www.geeksforgeeks.org/how-does-the-blockchain-work/>

The transaction process in a blockchain can be summarized as follows:

1. Facilitating a transaction: A new transaction enters the blockchain network. All the information that needs to be transmitted is doubly encrypted using public and private keys.
2. Verification of transaction: The transaction is then transmitted to the network of peer-to-peer computers distributed across the world. All the nodes on the network will check for the validity of the transaction like if a sufficient balance is available for carrying out the transaction.
3. Formation of a new block: In a typical blockchain network there are many nodes and many transactions get verified at a time. Once the transaction is verified and declared a legitimate transaction, it will be added to the mempool. All the verified transactions at a particular node form a mempool and such multiple mempools form a block.
4. Consensus Algorithm: The nodes that form a block will try to add the block to the blockchain network to make it permanent. But if every node is allowed to add blocks in this manner then it will disrupt the working of the blockchain network. To solve this problem, the nodes use a consensus mechanism to ensure that every new block that is added to the Blockchain is the one and only version of the truth that is agreed upon by all the nodes in the Blockchain, and only a valid block is securely attached to the blockchain. The node that is selected to add a block to the blockchain will get a reward and hence we call them “miners”. The consensus algorithm creates a hash code for that block which is required to add the block to the blockchain.
5. Addition of the new block to the blockchain: After the newly created block has got its hash value and is authenticated, now it is ready to be added to the blockchain. In every block, there is a hash value of the previous block and that is how the blocks are cryptographically linked to each other to form a blockchain. A new block gets added to the open end of the blockchain.
6. Transaction complete: As soon as the block is added to the blockchain the transaction is completed and the details of this transaction are permanently stored in the blockchain. Anyone can fetch the details of the transaction and confirm the transaction.

2.3.6.1 Proof of work vs proof of stake

Remember the idea of consensus mechanisms mentioned earlier? There are two ways blockchain nodes arrive at a consensus: through private blockchains, where trusted corporations are the gatekeepers of changes or additions to the blockchain, or through public, mass-market blockchains.

Most public blockchains arrive at consensus by either a proof-of-work or proof-of-stake system. In a proof-of-work system, the first node, or participant, to verify a new data addition or transaction on the digital ledger receives a certain number of tokens as a reward. To complete the verification process, the participant, or “miner,” must solve a cryptographic question. The first miner who solves the puzzle is awarded the tokens.

Originally, people on various blockchains mined as a hobby. But because this process is potentially lucrative, blockchain mining has been industrialized. These proof-of-work blockchain-mining pools have attracted attention for the amount of energy they consume.

In September 2022, Ethereum, an open-source cryptocurrency network, addressed concerns around energy usage by upgrading its software architecture to a proof-of-stake blockchain. Known simply as “the Merge,” this event is seen by cryptophiles as a banner moment in the history of blockchain. With proof-of-stake, investors deposit their crypto coins in a shared pool in exchange for the chance to earn tokens as a reward. In proof-of-stake systems, miners are scored based on the number of native protocol coins they have in their digital wallets and the length of time they have had them. The miner with the most coins at stake has a greater chance to be chosen to validate a transaction and receive a reward.

2.3.7 Why blockchain is important?

Business runs on information. The faster it’s received and the more accurate it is, the better. Blockchain is ideal for delivering that information because it provides immediate, shared and completely transparent information stored on an immutable ledger that can be accessed only by permissioned network members. A blockchain network can track orders, payments, accounts, production and much more. And because members share a single view of the truth, you can see all details of a transaction end to end, giving you greater confidence, as well as new efficiencies and opportunities.

2.3.8 Blockchain use cases

Blockchain's potential use cases span across industries, including financial services, retail, marketing and advertising, and healthcare. Here are some examples:

In financial services, blockchain increases settlement speed to real time (eliminating exchange rate risk for cross-currency transactions) and enables real-time transactions. It also has applications for simplifying operations, such as tracing bank guarantees and letters of credit across parties and executing smart contracts, making reporting faster and automating compliance.

Blockchain powers cryptocurrencies, which are digital currencies that are maintained by a decentralized system, resulting in cheaper and faster transactions.

Retailers are using NFTs, which are digital assets that sit on a blockchain, to engage with their tech-savvy customers and brand enthusiasts who want exclusive merch or experiences.

Luxury resale retailers are also using blockchain to certify the authenticity of their products and make the transfer of ownership more transparent.

In marketing, blockchain can be used to increase the security and transparency around the sharing of customer data, either between a customer and a company or between two companies.

Blockchain can also be used to reduce fraud and other trust-related issues in digital ad buying.

Blockchain has a wide range of applications in healthcare, including improving payment processing, electronic medical records, provider directories, and data security and exchange. Blockchain has been widely adopted for various applications, such as cryptocurrencies, smart contracts, supply chain management, digital identity, and product authentication. Product authentication is the process of verifying the origin, quality, and legitimacy of a product using various methods, such as labels, certificates, barcodes, RFID tags, or QR codes. Product authentication is important for consumers who want to ensure that they are buying genuine products that meet their expectations and standards. Product authentication is also important for manufacturers who want to protect their brand reputation, prevent counterfeiting, and increase customer loyalty.

2.3.9 Is blockchain secure?

In the most basic way, one can think of a blockchain as a linked list. Each of the next items in the list is dependent on the previous item, except for the first block, also known as the genesis block, which is hardcoded into the blockchain. In the blockchain, each block contains the hash of the previous block's header and a hash of the transactions in the Merkle tree of the current block. In this way, each block is cryptographically chained to the previous block. Let's understand with an example what happens when someone attempts to change a transaction or block data in a blockchain network.

Suppose, there is a chain of 10 blocks, where the 10th block depends on the 9th block, the 9th block depends on the 8th block, and so on.

In this way, the 10th block depends on all the previous blocks and the genesis block as well.

If someone tries to change data on the 2nd block, then the attacker will have to change data on all the later blocks as well, otherwise, the blockchain will become invalid since the later blocks depend on the hash value present in the 2nd block and the 2nd block has changed, but not the later blocks.

Thus, as the blocks are added, immutability increases as changing the block is an expensive operation.

Also, to add/change a block in a blockchain, a consensus algorithm is used by nodes in the blockchain network. In order to compensate for the change in one block, one must have to recalculate the hash of every block to update the hash value of the block header in the next block. This will involve a lot of time and computational resources.

In order to succeed with such kind of attack, the hacker has to simultaneously control and change 51% or more copies of the blockchain so that their new copy becomes the majority copy and thus the agreed-upon chain. Thus, requiring an immense amount of time, money, and computational resources.

2.3.10 Advantages of the blockchain

Numerous advantages of blockchain technology have been discussed in many industries and proposed by thought leaders around the world who are participating in the blockchain space. The

notable benefits of blockchain technology are as follows:

- **Decentralization:** This is a core concept and benefit of the blockchain. There is no need for a trusted third party or intermediary to validate transactions; instead, a consensus mechanism is used to agree on the validity of transactions.
- **Transparency and trust:** Because blockchains are shared and everyone can see what is on the blockchain, this allows the system to be transparent. As a result, trust is established. This is more relevant in scenarios such as the disbursement of funds or benefits where personal discretion in relation to selecting beneficiaries needs to be restricted.
- **Immutability:** Once the data has been written to the blockchain, it is extremely difficult to change it back. It is not genuinely immutable, but because changing data is so challenging and nearly impossible, this is seen as a benefit to maintaining an immutable ledger of transactions.
- **High availability:** As the system is based on thousands of nodes in a peer-to-peer network, and the data is replicated and updated on every node, the system becomes highly available. Even if some nodes leave the network or become inaccessible, the network as a whole continues to work, thus making it highly available. This redundancy results in high availability.
- **Highly secure:** All transactions on a blockchain are cryptographically secured and thus provide network integrity.
- **Simplification of current paradigms:** The current blockchain model in many industries, such as finance or health, is somewhat disorganized. In this model, multiple entities maintain their own databases and data sharing can become very difficult due to the disparate nature of the systems. However, as a blockchain can serve as a single shared ledger among many interested parties, this can result in simplifying the model by reducing the complexity of managing the separate systems maintained by each entity.
- **Faster dealings:** In the financial industry, especially in post-trade settlement functions, blockchain can play a vital role by enabling the quick settlement of trades. Blockchain does not require a lengthy process of verification, reconciliation, and clearance because a single version of agreed-upon data is already available on a shared ledger between financial organizations.

- **Cost saving:** As no trusted third party or clearing house is required in the blockchain model, this can massively eliminate overhead costs in the form of the fees which are paid to such parties.

2.3.11 Disadvantages of the blockchain

However, blockchain also faces some challenges for product authentication, such as:

- **Scalability:** Blockchain has limited scalability due to its decentralized nature and consensus mechanism. As the number of transactions and participants increases, the blockchain network may experience delays, congestion, and high fees. This may affect the performance and usability of the product authentication system.
- **Interoperability:** Blockchain has low interoperability with other systems and platforms due to its lack of standards and protocols. This may hinder the integration and communication of data across different supply chains and stakeholders.
- **Regulation:** Blockchain has uncertain regulation due to its novelty and complexity. There may be legal and ethical issues regarding data ownership, privacy, compliance, liability, and dispute resolution. These issues may vary depending on the jurisdiction and industry of the product authentication system.

2.4 QR Code for Product Authentication

2.4.1 What's QR Code?

Invented in 1994 by Masahiro Hara, chief engineer of Denso Wave, a Japanese company and subsidiary of Toyota, the QR code was initially used to track vehicles and parts as they moved through the manufacturing process.

Short for Quick Response, QR codes are a type of barcode easily readable with digital devices like smartphones. They store information as a series of pixels in a square grid that can be read in two directions top to bottom and right to left unlike standard barcodes that can only be read top to bottom.

QR codes can store about 7,000 digits or around 4,000 characters, including punctuation and special characters. It can also encode information like phone numbers or internet addresses. The arrangement of each QR code varies depending on the information it contains, and that changes the arrangement of its black modules.



Figure 5: QR Code versions

Source: https://en.wikipedia.org/wiki/QR_code

2.4.2 Static vs dynamic QR codes

QR codes vary in design depending on the encoded data and function, and can be categorized primarily in two ways: static and dynamic.

A static QR code cannot be modified once it has been created. This is ideal for creating QR codes in mass for an event. A drawback is its lack of creativity and that it may not allow for analytics on how many times the code may have been scanned. An example of a good static QR code would be one for your Wi-Fi password.

Dynamic QR codes allow you to change and edit the code as many times as you need. When the code is scanned, it redirects you to the URL contained inside. These codes offer the freedom to package your design, like adding contrasting colors. They also have the ability to track and measure advertising statistics.

These added insights allow the QR code creator access to where and with what device the code was scanned. Along with adding in campaign information and resetting scans, all the results collected can be downloaded as comma-separated values or a CSV report.

2.4.3 Usability of QR code

Nowadays, QR codes are still used to track products and product information through a supply chain, but they are also used for so much more. You've likely used a QR code to view a menu, link a social profile or add friends to an account, board a flight, download an app, send and receive payments, access Wi-Fi, and authenticate your login details. The possibilities with QR codes are truly endless.

2.4.4 QR code components

Visually, a QR code looks like a twisted crossword puzzle, but its design is crucial to its function. Here are some of its most important elements.

Position detection markers: The prominent squares located in three corners of each code offer easier recognition and assist with reading the QR code at high speed.

Alignment markers: These help straighten out codes placed on curved surfaces. It's smaller than a position detection marker but will become larger the more information a QR code holds.

Timing pattern: The black and white alternating modules configure the data grid and help the scanner calculate how large the data matrix is.

Version information: This determines which of the 40 different QR code versions is being used, with the most common versions being 1 to 7.

Format information: This pattern holds information about the data mask pattern and error tolerance of the code, making it easier to scan.

Data and error correction keys: The error correction function shares a structural space where all the data in a QR code is contained. This correction block's mechanism is essential to allowing up to 30% of a code being read if damaged.

Quiet Zone: This white space can be seen as the border of a QR code to help improve comprehension for scanning and provide structure. It determines what is and isn't part of the code.

2.4.5 Advantages and disadvantages of QR Codes

QR codes are becoming increasingly popular as a way to store and transfer information. QR stands for “Quick Response” and is a type of two-dimensional barcode. QR codes are becoming popular in marketing, advertising, and other industries due to their ability to store large amounts of data in a small space. However, there are both advantages and disadvantages to using QR codes. In this article, we will discuss the advantages and disadvantages of QR codes.

2.4.5.1 Advantages of QR Codes

Easy to Create and Use: One of the main advantages of QR codes is that they are easy to create and use. All you need is a computer and an internet connection to create a QR code. Once created, the code can be scanned with a smartphone or other device to access the information stored in the code. This makes QR codes an ideal solution for quickly and easily sharing information.

Cost-Effective: Another advantage of QR codes is that they are cost-effective. QR codes can be created and used for free, which makes them an attractive option for businesses and organizations

looking to save money.

Versatile: QR codes are also versatile. They can be used to store a variety of different types of information, including text, images, and links. This makes them a great option for businesses and organizations that need to store and share a lot of information.

2.4.5.2 Disadvantages of QR Codes

Limited Reach: One of the main disadvantages of QR codes is that they have a limited reach. QR codes can only be scanned by devices that have the necessary software installed. This means that not everyone will be able to access the information stored in the code.

Security Concerns: Another disadvantage of QR codes is that they can pose a security risk. Since QR codes can store a lot of information, it is possible for someone to access the code and gain access to sensitive information.

Prone to Damage: QR codes are also prone to damage. If the code is exposed to water or other elements, it can become unreadable. This can make it difficult to access the information stored in the code.

CHAP III. DESIGN AND IMPLEMENTATION

3.1 Requirements

As I earlier introduced already this product authentication system baptized “ProAuth” based on web and mobile applications using mainly Blockchain and secure QR Code is a system that aims to verify the legitimacy and traceability of products by using a combination of various technologies. Such a system can benefit both consumers and producers by ensuring the accuracy of labeling, originality, preventing counterfeit goods and informing customers about the authenticity of the product.

After choosing this subject, I had to think about and list out all features and functionalities that were going to be included in this system.

As the main actors were manufacturers of the products and the customers, I split all features in two groups. The first group was including all features related to manufacturers who are the owners of products, brands. And the second group was including all features related to customers. By doing this. It help me to think about the appropriate tools and approaches to take into account by implementing these features in a most convenient and reliable way.

After listing out all the features that I needed in the system, the next main challenge was to elaborate and chose the tools and technologies that I was going to be using during the implementation of this system. I did plenty researches about new technologies, understand how they work, the advantages they give on a high level implementation, the disadvantages and I tried to balance the pros and cons of each one.

In a nutshell, the design and implementation of the proposed system has involved the following steps:

- Define the scope and objectives of the system, such as what kind of products, data, and stakeholders are involved, and what are the expected outcome and challenges.
- Design and develop a web application and a mobile application that can interact with the blockchain network, database and provide a user-friendly interface for scanning, verifying, and tracking products using secure QR codes.

- Choose a suitable blockchain platform and framework, such as Ethereum that can support the system's requirements and functionalities, such as smart contracts, consensus mechanisms, and scalability.
- Design and develop a RESTful web service that can communicate with the database and the applications. Provide a standardized and consistent way of exchanging data and requests.
- Design and develop a database that can store, manage the data and records of the products such as product unique identification, blockchain hashes, and metadata. The database should also be synchronized with the blockchain network and the web service, and support queries and analytics.
- Test and evaluate the system's performance, usability, reliability and identify any issues or improvements that can be made.

In the pursuit of developing a robust and efficient application, I have meticulously designed a high-level diagram that encapsulates the architectural blueprint and core functionalities of the proposed system. This pivotal diagram serves as a visual representation of the application's structure, providing a comprehensive overview of its components and interactions.

The significance of the designed high-level diagram lies in its ability to elucidate the intricate design aspects, offering stakeholders, developers, and readers a concise yet thorough understanding of the application's architecture and flow.

First and foremost, the diagram acts as a foundational roadmap, delineating the major components, modules, and their interconnections within the system. It enables a holistic view, highlighting how various elements harmoniously converge to fulfill the application's intended purpose. It makes very easy and clear the understanding of the application's architecture and flow

Moreover, the diagram serves as a communication tool, bridging the gap between technical intricacies and comprehensibility. Its visual nature simplifies complex technical details and assists in maintaining a clear focus on design objectives, ensuring adherence to the defined architecture.

The incorporation of this comprehensive high-level diagram aligns with the thesis's endeavor to present a robust and well-structured approach towards application design, ultimately contributing

to the successful realization of the envisioned system.

Let's see the overall high level architecture and I will be following up with the detailed information about the technology stacks that I have used.

3.2 High level architecture diagram

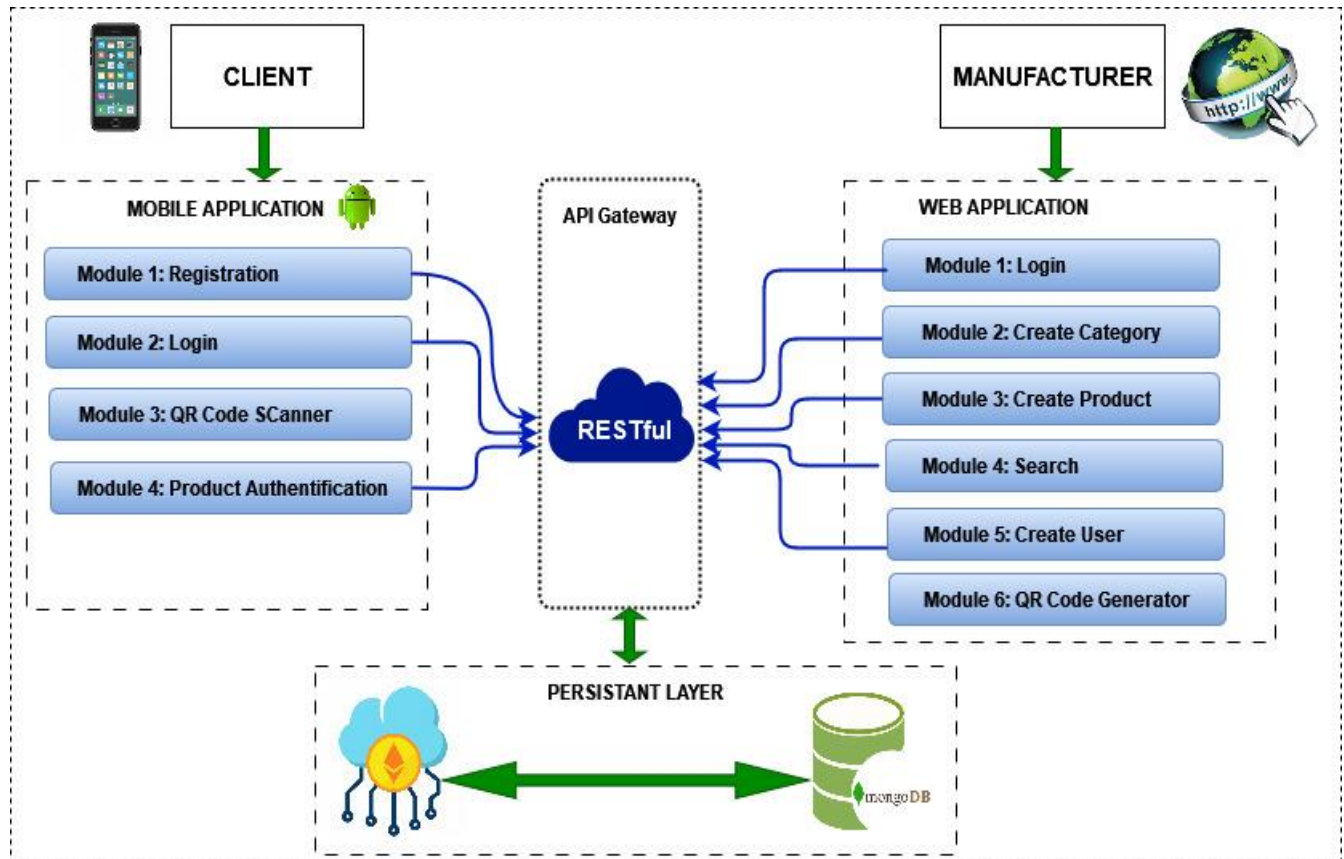


Figure 6: High level architecture diagram

As it's presented in the high level architecture diagram, the ecosystem has four main components which are as follows:

- Web application
- Mobile application
- API Gateway
- Persistent layer

3.2.1 Web application

The web application will be a tool to be used by manufacturers of the various product. It's the interface between the manufacturer and the persistent layer. It will be beneficial for manufacturers to manage their products such as creating the categories of the products, storing information of their products in the blockchain and having another copy in the database. This will give the advantage of immutability as once recorded on the blockchain, the data becomes immutable. It cannot be altered or deleted without consensus from the network, ensuring data integrity and eliminating the need for a central authority. This decentralization helps in preventing a single point of failure and improves reliability. It allows as well to retrieve products and search whatever product by id, reference or name.

The web application gives the possibility to create a user and allocate an access right based on the position or role of the user. It allows to view the reports of all components managed in the system such total count of all the products already created and so on.

The last thing but not the least it allows to generate a QR code of the product with a unique digital identifier.

3.2.2 Mobile application

The mobile application is an Android based application. It serves as interface between the customer which is the consumer and the database where product information are stored. It allows the customer to submit their requests about authenticating a product to the depot of data. It gives the customer a feature to scan a product QR Code and get back the conformation whether the product is original or fake. It gives the consumers as well the possibility to register themselves before using the application and login before accessing the features with the application.

3.2.3 API Gateway

The API Gateway plays an important role here as it allows to decouple the system and make possible to extend it easily in the future without touching all unnecessary modules. It is a RESTful web service based and it serves an interface between first the web application and the persistent layer and second the mobile application and persistent layer. It's taking care all requests from the

web and the mobile applications, submits all of them to the persistent layer and revert back with the appropriate response to the initiator source.

For example, if a manufacturer wants to create a new category of the product, the user interacts with the web application's interface by completing all information related to the new category. The API Gateway receives the incoming request from the web application. It performs necessary validations, security checks, and routing. After processing the request, the API Gateway forwards the validated request to the persistent layer, often through an API endpoint associated with category creation. The persistent layer, which includes the database system, receives the request from the API Gateway. It executes operations to create a new category record in the database. The database system ensures data integrity and performs any additional operations, such as indexing or ensuring uniqueness constraints. Once the category creation process is completed in the database, the persistent layer sends a response or status update indicating the success or failure of the operation back to the API Gateway. Based on the response received from the persistent layer, the API Gateway formulates a response to the initial request from the web application. The API Gateway sends this response back to the web application, indicating whether the category creation was successful or encountered an error. The web application processes the response received from the API Gateway. If the operation was successful, the web application displays a successful message to the user. In case of a failure, appropriate error handling mechanisms in the web application handle the issue, notifying the user of the problem encountered.

3.2.4 Persistent layer

The persistent layer is another important component here. It's made up with Blockchain based on Ethereum and a database of MongoDB version. Both are used to store information within this ecosystem.

Blockchain is used to store products securely in a decentralized manner and the MongoDB database is responsible of storing a copy of the blockchain information with the signature hash of each product in the blockchain. It stores as well all remaining information in different tables.

3.2.4.1 Why combining Blockchain and MongoDB ?

Combining these two tools offer various advantages as following:

Data Integrity and Security: Blockchain's immutability complements MongoDB's robust security measures, ensuring data stored in MongoDB remains secure and unaltered.

Scalable and Flexible Architecture: Using MongoDB alongside Blockchain allows for a scalable and flexible architecture. MongoDB's scalability supports the growth of data, while Blockchain's decentralized nature adds security and integrity.

Versatile Application Scenarios: The combination of Blockchain and MongoDB can cater to diverse application scenarios, leveraging the strengths of both technologies for use cases demanding security, transparency, and data flexibility.

Integrating MongoDB as a flexible and efficient database layer with the security and immutability of Blockchain can create powerful, scalable, and secure systems suited for various applications requiring data integrity and robustness.

3.3 Sequence diagram

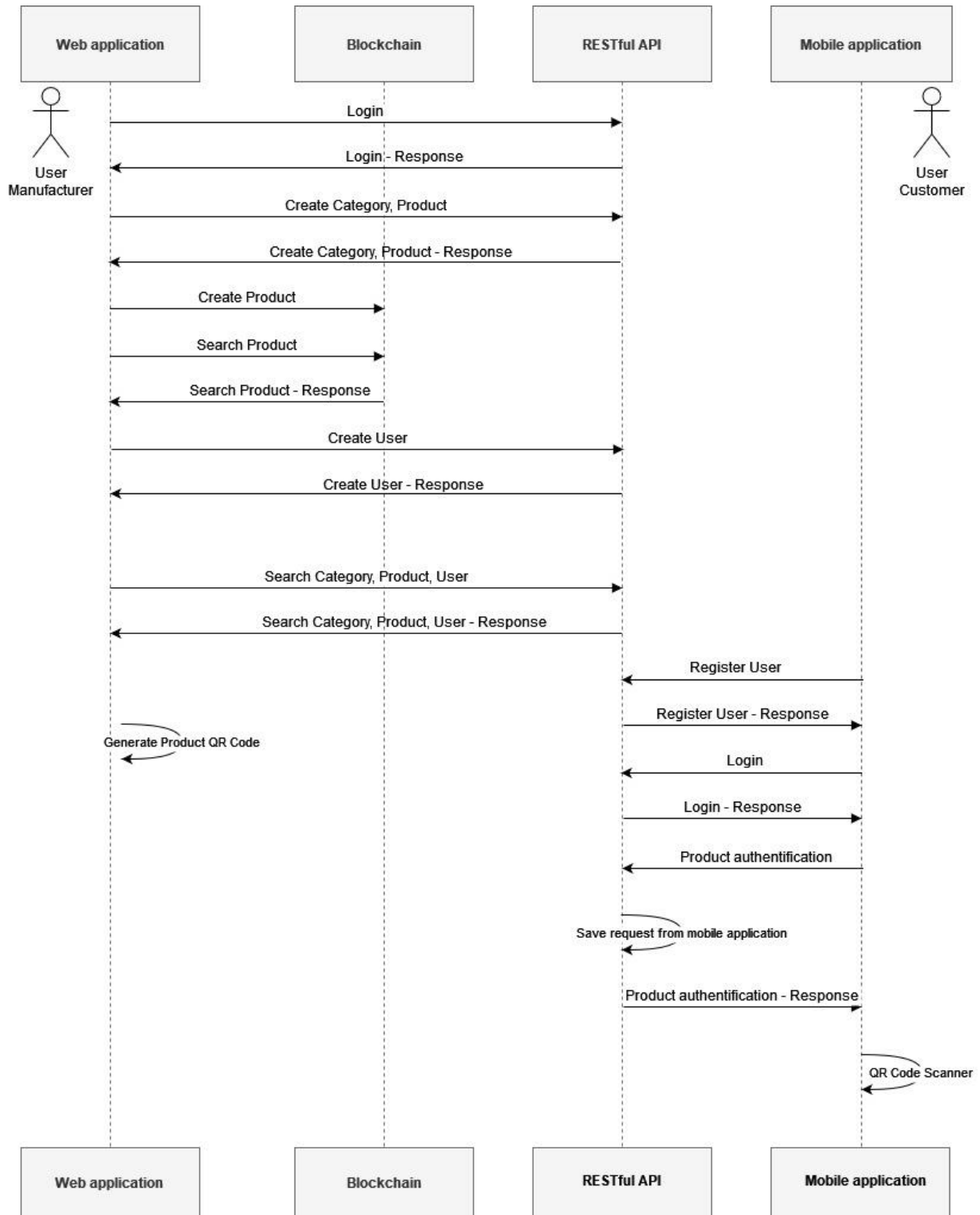


Figure 7: Sequence diagram

3.4 Class Diagrams

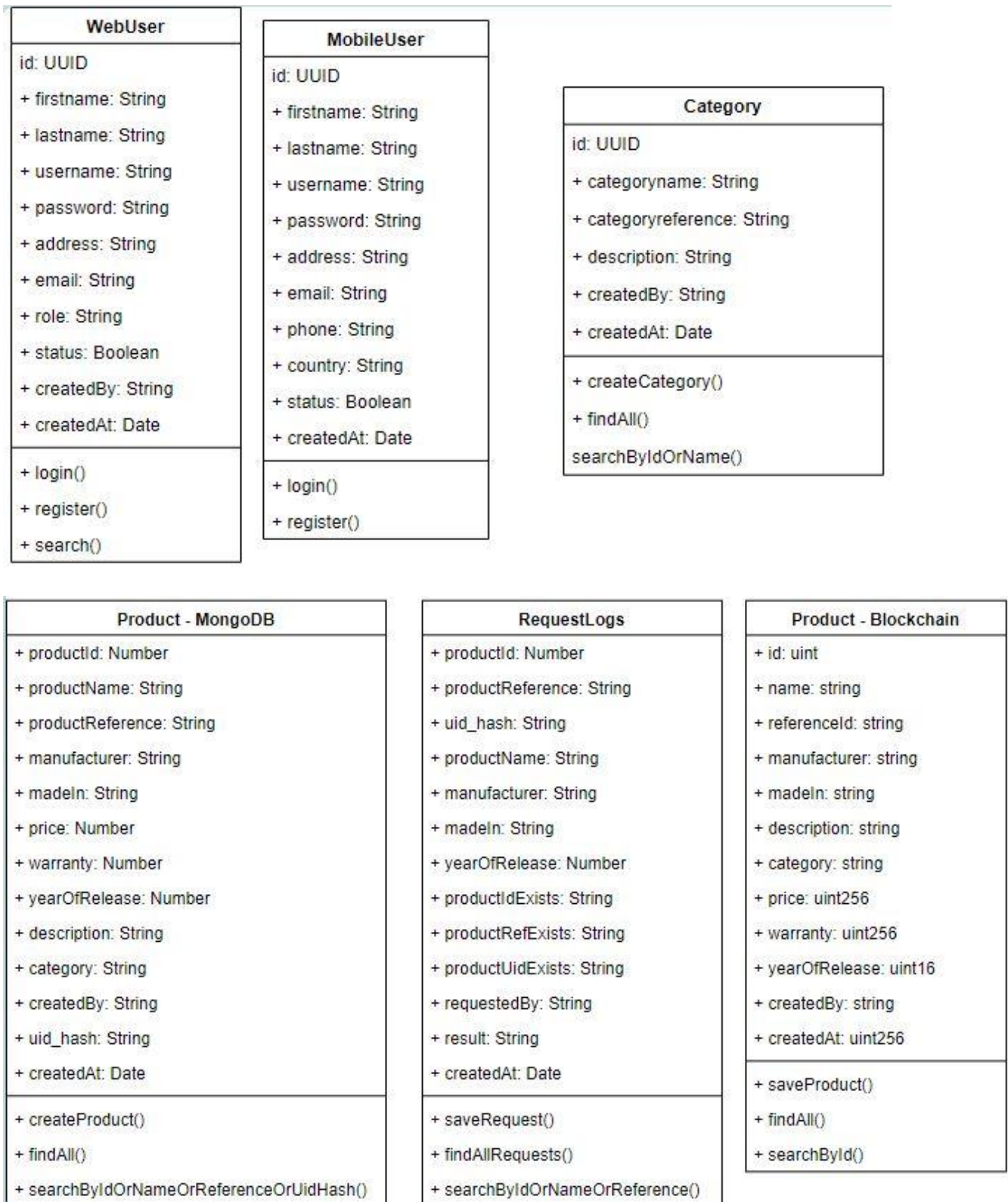


Figure 8: Class diagrams

3.5 Involved technologies and tools

As the high level diagram showed, we have four main components in this ecosystem and we are trying to list out all involved technology stacks per each component.

3.5.1 Web application technology stacks

HTML5 : stands for HyperText Markup Language version 5, is the latest iteration of the standard language used for structuring and presenting content on the World Wide Web. It's the cornerstone technology for creating web pages and web applications, providing a standardized system for structuring content, adding multimedia elements, and building web-based software.

CSS3: stands for Cascading Style Sheets level 3, is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS3 is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

JavaScript: is a scripting or programming language that allows you to implement complex features on web pages. It is used to create interactive effects within web browsers. JavaScript can update and change both HTML and CSS. It can calculate, manipulate, and validate data. JavaScript is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS). JavaScript is a language of style rules that we use to apply styling to our HTML content, for example setting background colors and fonts, and laying out our content in multiple columns.

NodeJS: is a JavaScript runtime environment that allows you to run JavaScript code outside of a web browser. It is built on browser's JavaScript engine and provides an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js is used for building scalable network

applications, such as web servers. It is also used for building command-line tools and desktop applications. Node.js is an open-source, cross-platform, and free to use.

Bootstrap: is a popular CSS framework that allows developers to create responsive and mobile-first websites. It is an open-source framework that provides a set of pre-built CSS classes and JavaScript plugins that can be used to create responsive web pages. Bootstrap is designed to make it easy to create responsive web pages that work well on all devices, including desktops, tablets, and smartphones. It includes a grid system that allows developers to create responsive layouts, as well as a variety of pre-built components such as navigation bars, forms, buttons, and more. Bootstrap is widely used by web developers and is supported by a large community of developers who contribute to its development.

3.5.2 Mobile application technology stacks

Java: is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA). Java is widely used for developing mobile applications, web applications, and games. It is also the first programming language and development platform for enterprise applications, with millions of developers running more than 60 billion Java Virtual Machines worldwide.

XML: stands for eXtensible Markup Language. It is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. Unlike HTML, XML does not have predefined tags to use. Instead, you define your own tags designed specifically for your needs. This makes it a powerful way to store data in a format that can be stored, searched, and shared. XML is often used in web development, data storage, and data exchange between applications.

Gradle: is a build automation tool that helps developers build, automate, and deliver better software, faster. It is a free and open-source tool that supports multiple languages such as Java, C++, and Kotlin. Gradle is known for its performance, flexibility, and ease of use. It is widely used in the software development industry to manage dependencies, compile code, and run tests.

3.5.3 API Gateway technology stacks

Express: is a web application framework for Node.js that provides a set of features for building web and mobile applications. It is used to build single-page, multipage, and hybrid web applications. Express is built on top of Node.js and helps manage servers and routes. It also includes a number of middleware modules that can be used to execute additional requests and responses activities. Express is simple to set up and personalize, and it allows you to define application routes using HTTP methods and URLs. It is also simple to interface with a variety of template engines, including Jade, Vash, and EJS.

Postman: is a standalone software testing API platform that allows you to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses. Postman enables you to store, catalog, and collaborate around all your API artifacts on one central platform.

3.5.4 Persistent layer technology stacks

Ethereum blockchain: is a decentralized blockchain technology that is not owned or regulated by a third party such as a government or central bank. It is used for building decentralized apps (dApps), holding and transacting cryptocurrency and other digital assets, and creating new cryptocurrencies. Ethereum is a network of computers all over the world that follow a set of rules called the Ethereum protocol. The Ethereum network acts as the foundation for communities, applications, organizations and digital assets that anyone can build and use. In a blockchain, transactions are recorded in a database that is updated and shared across many computers in a network. Every time a new set of transactions is added, it is called a “block” hence the name blockchain. Public blockchains like Ethereum allow anyone to add, but not remove, data. If someone wanted to alter any of the information or cheat the system, they would need to do so on the majority of computers on the network.

Ganache: is a personal blockchain for Ethereum development that can be used for testing smart contracts and DApps in a sandbox environment. It allows developers to create and manage a personal Ethereum blockchain for testing and development purposes. Ganache can simulate different network conditions, such as network latency and limited bandwidth, to test the performance of the smart contracts and DApps. It is developed by Truffle Suite and comes in two

versions: a desktop application with a user interface (Ganache UI) and a command-line tool (Ganache CLI).

MetaMask: is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications. MetaMask is the leading self-custodial wallet that provides a safe and simple way to access blockchain applications and web3. It is trusted by millions of users worldwide. Developers can contribute to MetaMask, which is powered by a strong community from across the globe.

Web3: is a collection of libraries that allow you to interact with a local or remote Ethereum node using HTTP, IPC or Web Socket. It provides a simple and easy-to-use API for developers to interact with the Ethereum blockchain and smart contracts deployed on the blockchain.

Truffle: is a comprehensive development framework and toolset for building decentralized applications (dapps) on the Ethereum blockchain. It offers a range of powerful tools that streamline the entire development process, from smart contract creation and testing to deployment and asset management. Truffle provides a suite of tools that includes a development environment, testing framework, and asset pipeline. It also offers a built-in smart contract compilation, linking, deployment, and binary management.

Solidity: Solidity is a programming language used for creating smart contracts on the Ethereum blockchain. It is a high-level language that is object-oriented and contract-oriented. Solidity is designed to be secure, reliable, and easy to use. It is influenced by Python, C++, and JavaScript. Solidity is the primary language for creating smart contracts on the Ethereum blockchain.

MongoDB: is a general-purpose, open-source, document-oriented database program that is designed to store and manage large volumes of data. It is categorized under the NoSQL (Not only SQL) database because the storage and retrieval of data in MongoDB are not in the form of tables. MongoDB stores data in flexible, JSON-like documents that map to the objects in your application code. It is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use. MongoDB is free to use.

Mongoose: is a MongoDB object modeling tool that provides a simple and elegant way to interact with MongoDB databases. It is designed to work in an asynchronous environment and provides a schema-based solution to model application data and supports basic validation. Mongoose also provides a uniform API for accessing numerous different databases, including Redis, MySQL, LDAP, MongoDB, and Postgres.

3.6 Implementation

This time, we'll embark on a comprehensive exploration of our product authentication system's implementation. Our focus will be on two key applications: a web application tailored for manufacturers and a mobile application designed for customers seeking to validate product authenticity.

The web application serves as a robust platform empowering manufacturers. It enables them to efficiently manage and store all useful product details leveraging the innovative capabilities of blockchain technology. On the other hand, the mobile application caters to the convenience of customers. By simply scanning a QR code, they can swiftly authenticate a product's legitimacy by checking if it's a real or a fake one.

We'll delve deeply into the intricacies of both applications, dissecting their features and functionalities. Rich visual aids, including images and descriptive annotations, will guide us through these applications' user interfaces. These visuals will illuminate each feature, offering a vivid explanation of how the apps operate and interact.

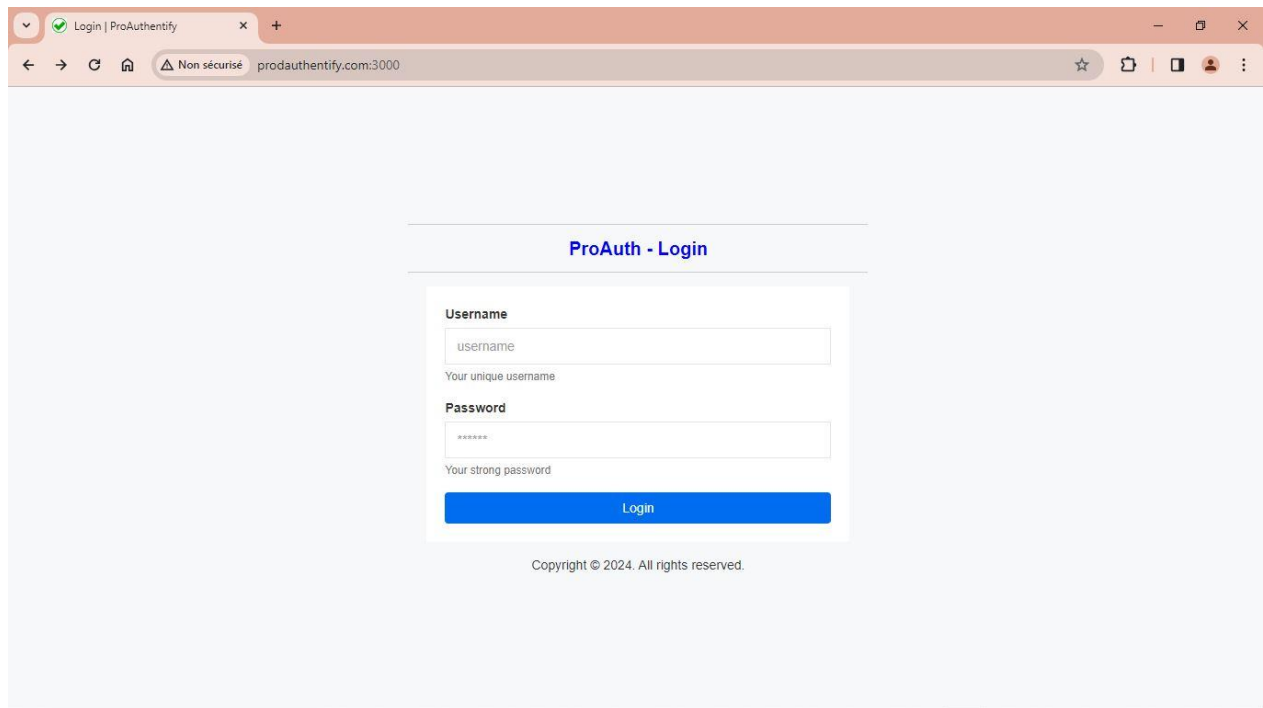
This chapter aims to elucidate how these interconnected applications synchronize, illustrating a seamless synergy between manufacturers and customers in ensuring product authenticity. Through detailed descriptions and visual representations, we endeavor to illuminate the practicality and significance of our system's implementation. We'll take a look at pictures of what these applications look like and explain what each part does.

3.6.1 Web application

3.6.1.1 Login

Starting with the Login process, the manufacturer users have to authenticate themselves with a valid username and password. When both credentials are corrects they are redirected to the homepage containing all features embedded in the web applications.

By requiring login credentials, the system regulates access, ensuring that only authorized users created by manufacturers can enter and engage with the platform. This serves as a pivotal security measure, protecting sensitive information and functionalities.

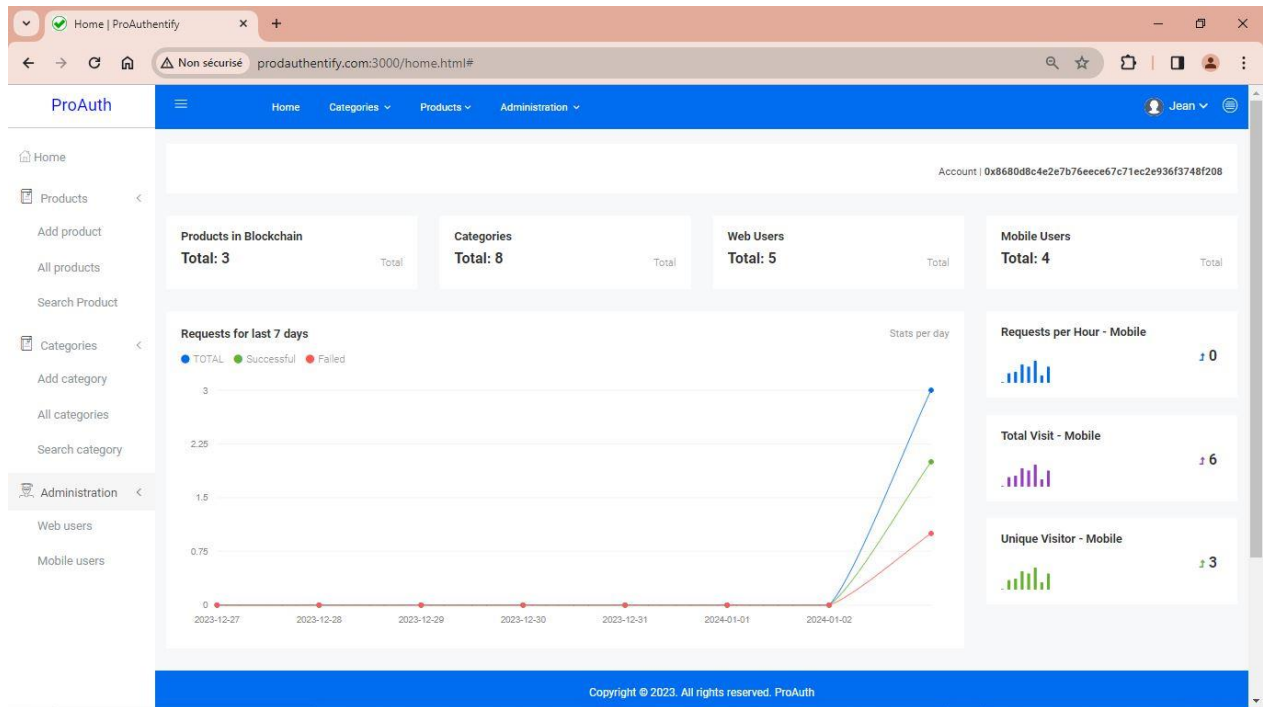


The screenshot shows a web browser window with the title 'Login | ProAuthentify'. The address bar displays 'Non sécurisé prodauthentify.com:3000'. The main content area features a login form titled 'ProAuth - Login'. The form includes a 'Username' field with the placeholder text 'username' and a subtext 'Your unique username'. Below it is a 'Password' field with a masked password '*****' and a subtext 'Your strong password'. A blue 'Login' button is positioned at the bottom of the form. At the very bottom of the page, a copyright notice reads 'Copyright © 2024. All rights reserved.'

Login - Web app

3.6.1.2 Home

When the login step is successful, the user in session is redirected to the home page which contains the following menus for managing products, categories, user administrations. In addition, it contains various statistics as total number of products stored in blockchain, categories, web and mobiles users, mobile requests per hour, mobile visits and unique visitors.



Home - Web app

3.6.1.3 Products

Product menu has three sub-menus that allow to view, add and search the products.

Add Product | ProAuthify

Non sécurisé prodauthify.com:3000/add-product.html

ProAuth

Home Categories Products Administration Jean

New Product All Products Search Product

Product name...

Year of release...

Product reference...

Description

Manufacturer...

Made In Country...

Price...

Warranty...

Select Category

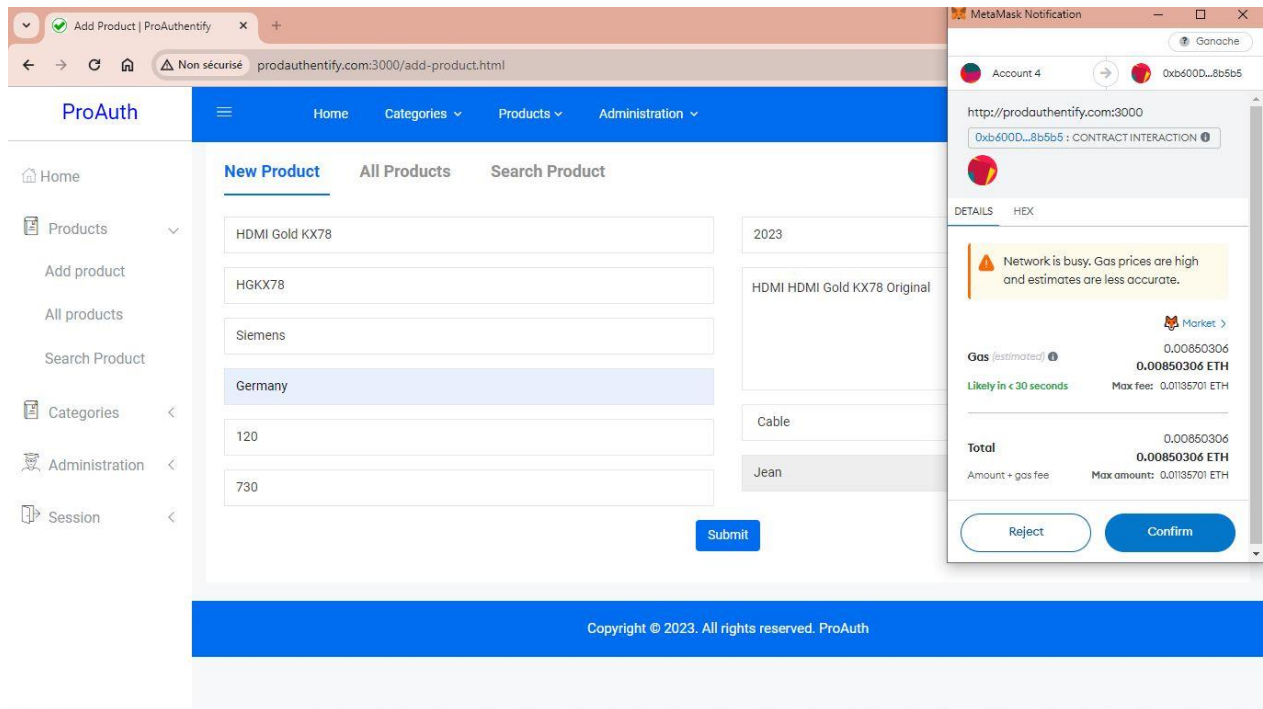
Jean

Submit

Copyright © 2023. All rights reserved. ProAuth

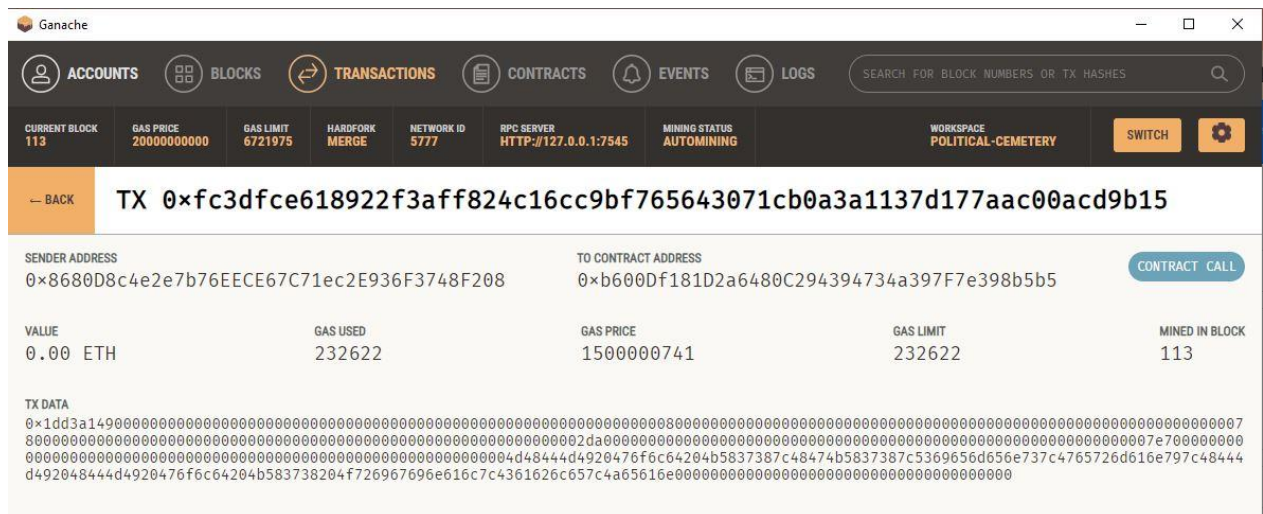
Create a new product - Web app

During the step of creating a new product, the user has to confirm the transaction in blockchain within metamask.



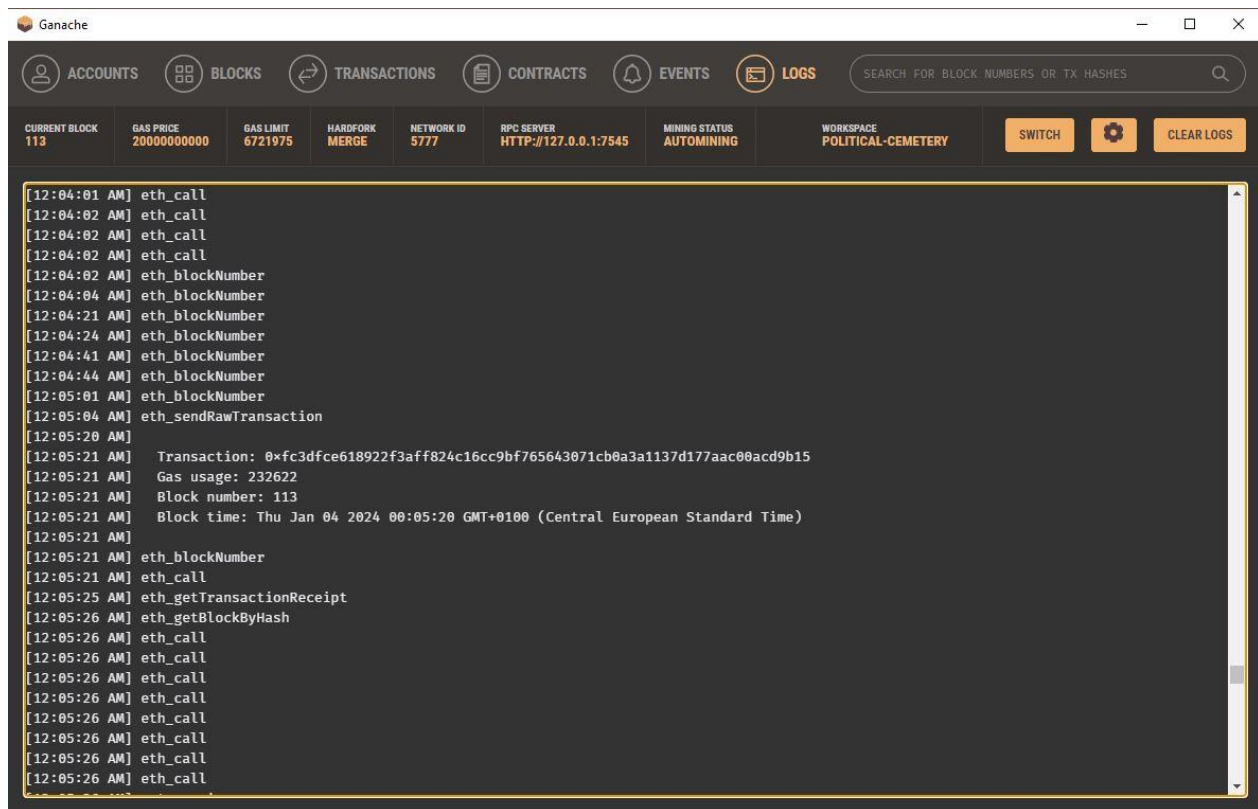
Create new product - Web app

After user confirmation, the operation is completed and details can be seen within Ganache tool under menu “TRANSACTIONS”



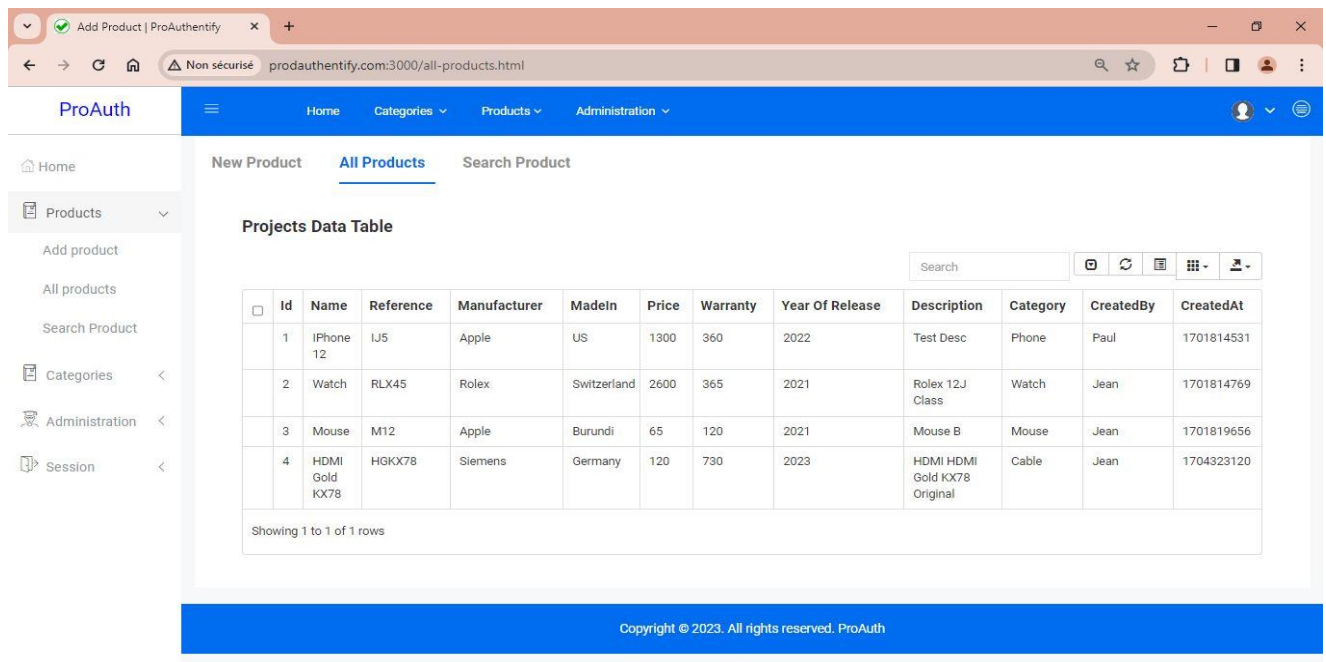
Ganache transactions - interface

The same can be checked and verified as well under Ganache “LOGS”



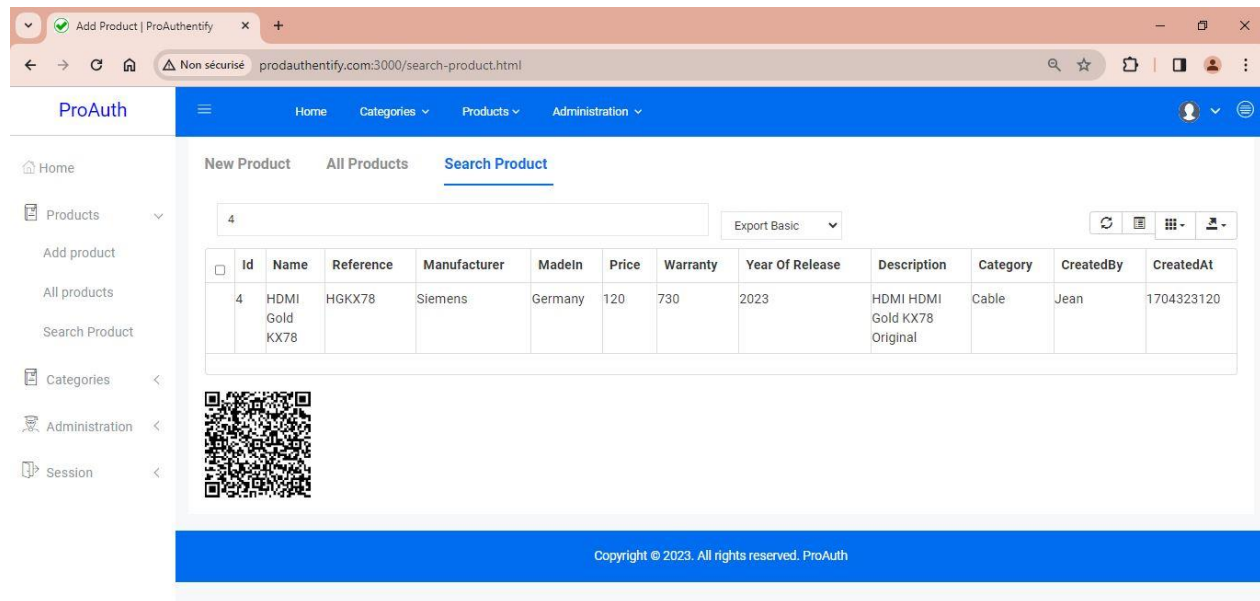
Ganache logs – interface

The user can visualize all products stored in Blockchain.



View all products – Web app

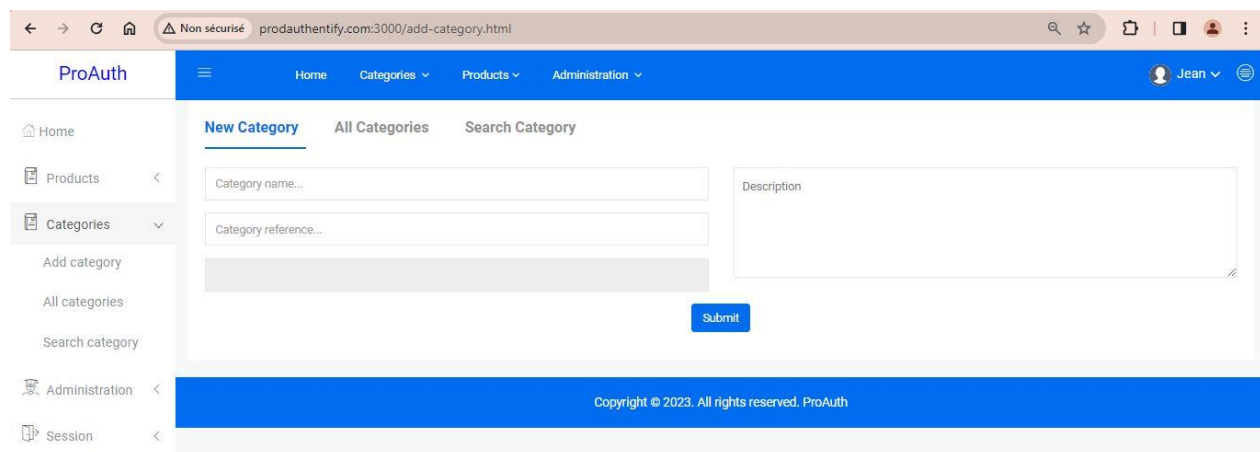
The user can search any product by identification number and view all relevant information with the associated QR Code.



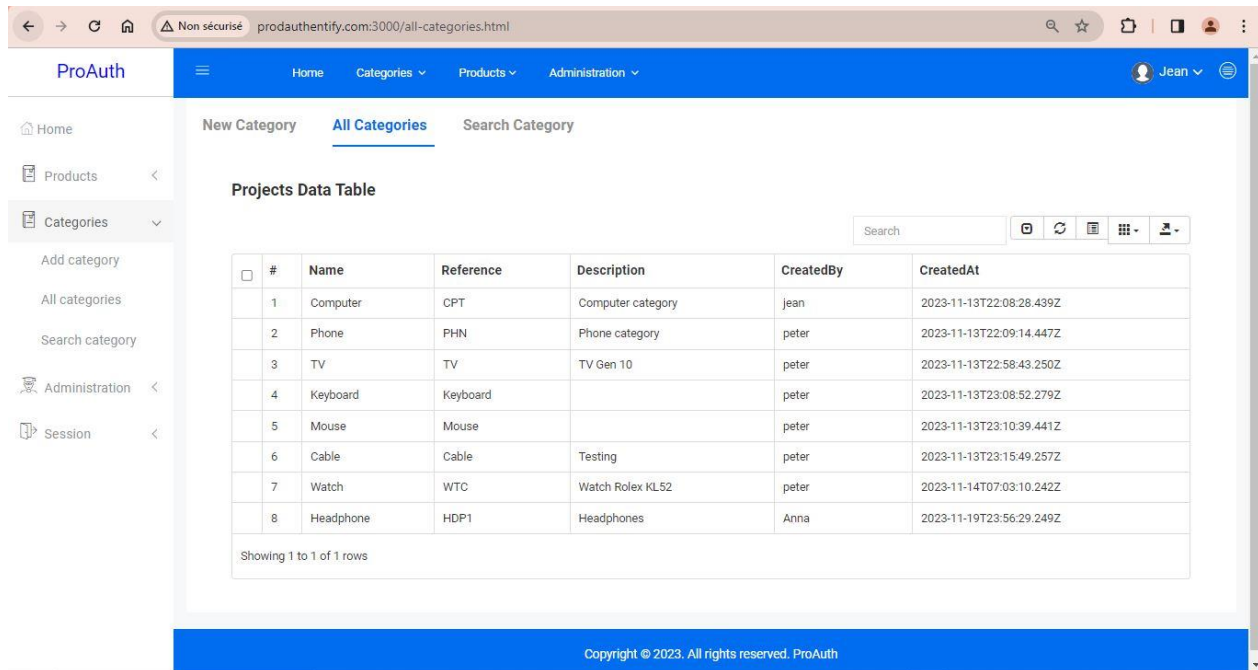
Search product - Web app

3.6.1.4 Categories

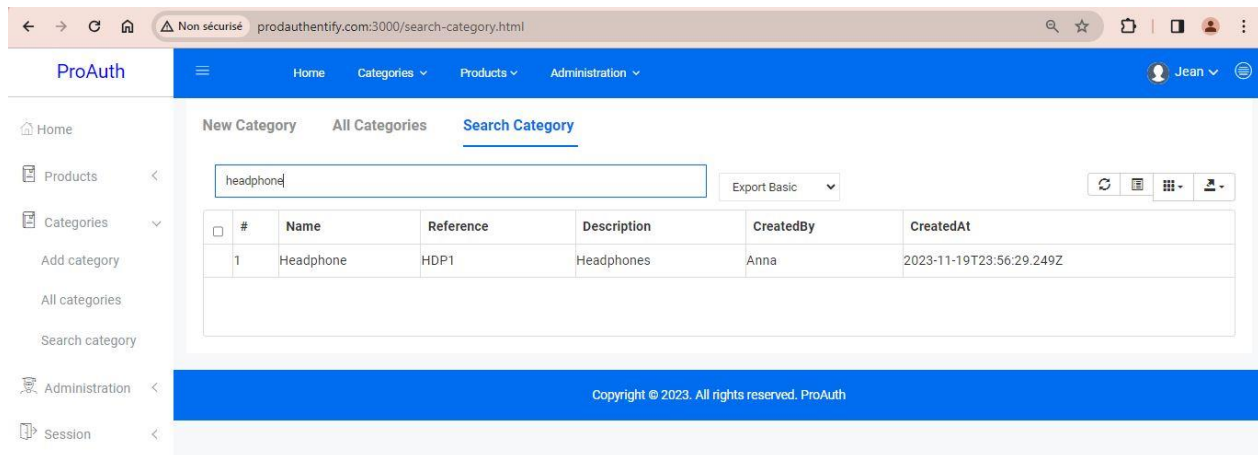
As each product is categorized, a user can create a new category of the product. The system is flexible to store different categories of products, thus there's a menu dedicated to manage categories within the system. Besides creating a new category, there's a possibility as well to view all available categories and search a specific category by name.



Create new category - Web app



View all categories - Web app



Search category - Web app

3.6.1.5 Administration

3.6.1.5.1 Web users

The user with administration right can create a new user of the web application, view all users and search a user by username.

ProAuth

Home Categories Products Administration

Home

Products

Categories

Administration

Web users

Mobile users

Session

New W-User All W-Users Search W-User

Firstname... Address...

Lastname... Email...

Username... Role_Standard

Password... Jean

Submit

Copyright © 2023. All rights reserved. ProAuth

Add web user - Web app

ProAuth

Home Categories Products Administration

Home

Products

Categories

Administration

Web users

Mobile users

Session

New W-User All W-Users Search W-User

Projects Data Table

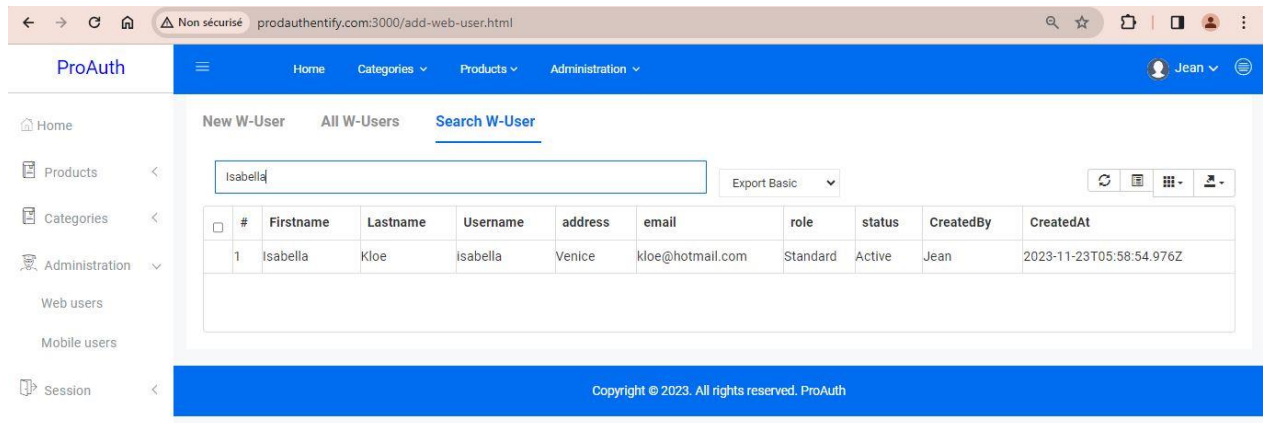
Search

	#	Firstname	Lastname	Username	address	email	role	status	CreatedBy	CreatedAt
<input type="checkbox"/>	1	Peter	Klein	Peter	Berlin	peter@outlook.com	Standard	Inactive	Jean	2023-11-15T06:42:46.815Z
<input type="checkbox"/>	2	Jean	Helmes	Jean	New York	jean@outlook.com	Admin	Inactive	Peter	2023-11-15T06:43:34.530Z
<input type="checkbox"/>	3	Johon	Gallagher	John	Paris	john@gmail.com	Standard	Inactive	jean	2023-11-15T07:35:49.437Z
<input type="checkbox"/>	4	Anna	Klouse	Anna	Warsaw	anna@yahoo.com	Admin	Inactive	jean	2023-11-15T08:01:32.215Z
<input type="checkbox"/>	5	Isabella	Kloe	isabella	Venice	kloe@hotmail.com	Standard	Active	Jean	2023-11-23T05:58:54.976Z

Showing 1 to 1 of 1 rows

Copyright © 2023. All rights reserved. ProAuth

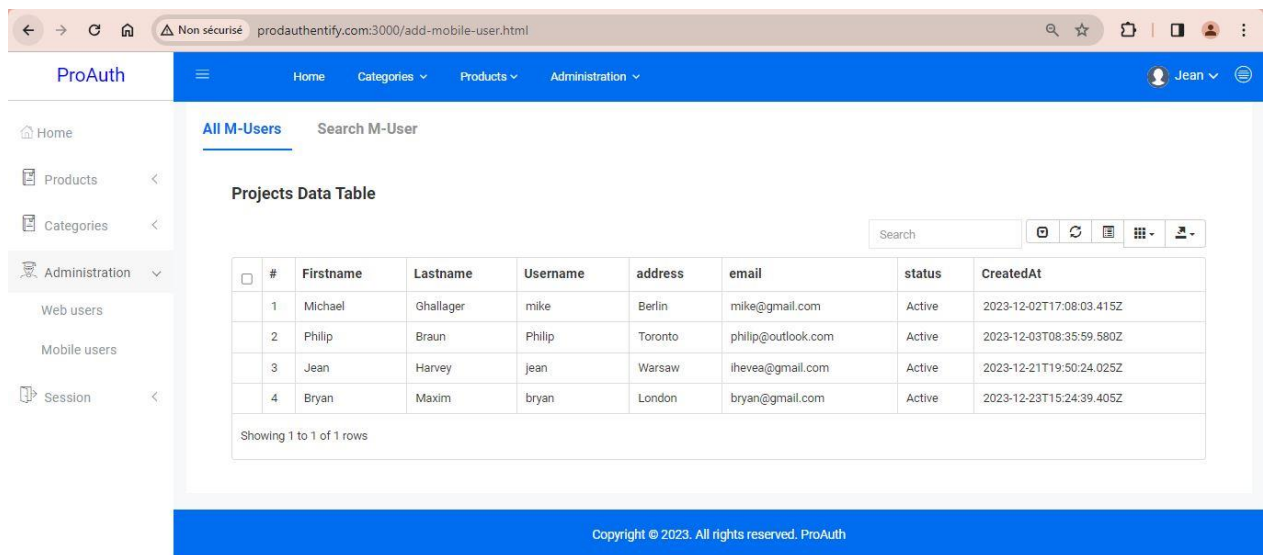
View web user - Web app



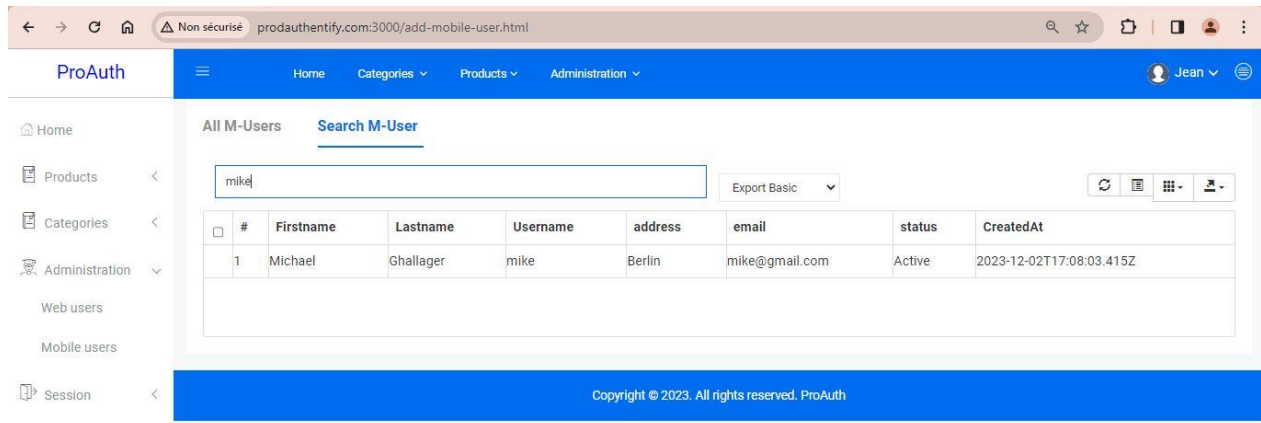
Search web user - Web app

3.6.1.5.2 Mobile users

The user with administration rights can view as well all users of the mobile application and search a specific user by username.



View mobile users - Web app



Search mobile users - Web app

3.6.1.6 Session

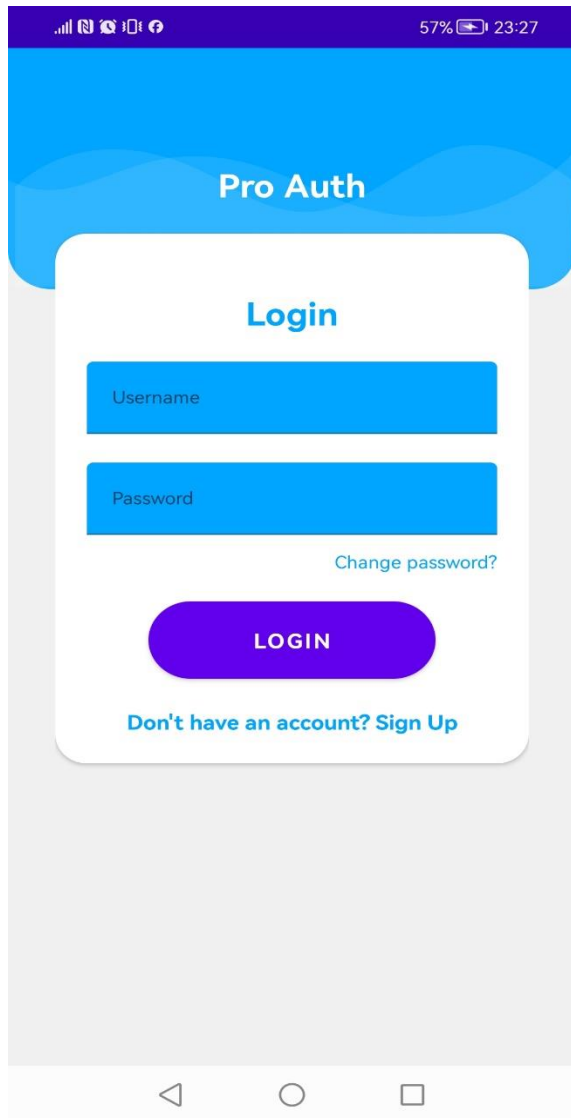
The session menu is helping the user to logout securely in the application.

3.6.2 Mobile application

3.6.2.1 Login

It provides a secure entry point, allowing only authorized users to access the application's features and functionalities. The user needs to provide the valid username and password first to be able to use the functionalities included in the application.

Besides the login components, it provides an option to go to sign up if the user does not have an account yet.



Login - Mobile app

3.6.2.2 Registration

The registration process is a prerequisite for customers before using the mobile application. We'll delve into the benefits inherent in this essential step. Registration empowers customers to create individual profiles, enabling a tailored and personalized experience within the application. This personalization often includes useful data as customer name, email which may be used later for sending notifications to the customer and some various settings curated to enhance user interactions. The user has a possibility to activate or not the notification setting.

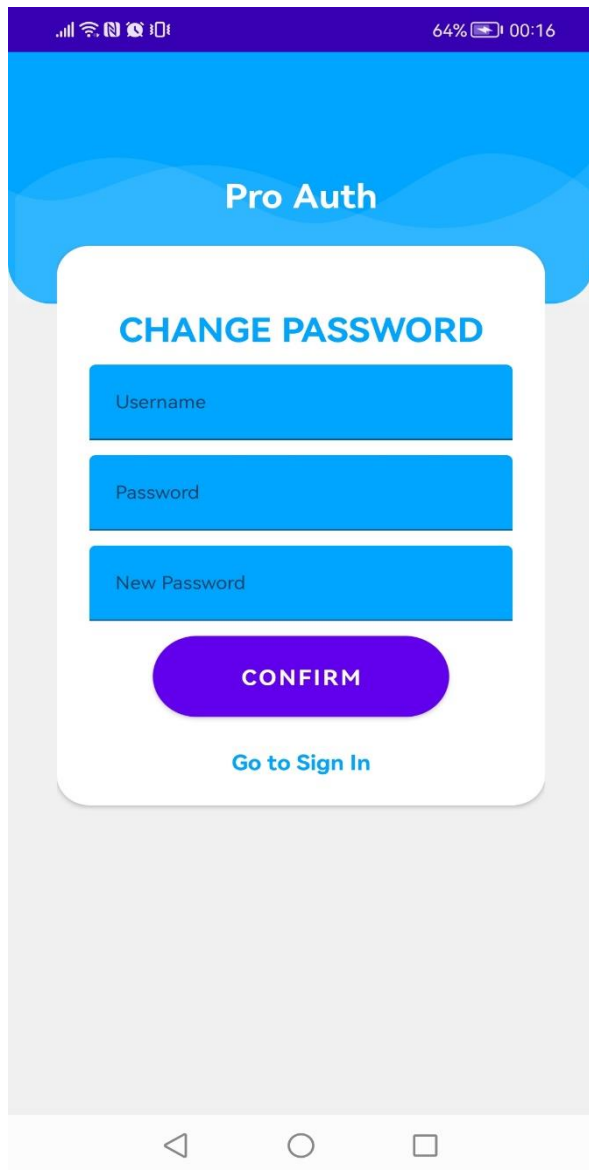
The screenshot shows a mobile application interface for registration. At the top, a status bar displays signal strength, battery level at 57%, and time at 23:28. Below this is a blue header with the text 'Pro Auth'. The main content area is white and contains the title 'REGISTER' in blue. There are eight input fields, each with a blue background and white text: 'First name', 'Last name', 'Username', 'Password', 'Address', 'Email', 'Phone number', and 'Country' (which has a dropdown arrow). Below the input fields is a toggle switch for 'Email notification', which is currently turned on. At the bottom of the form is a large, rounded blue button with the text 'REGISTER' in white. The entire app interface is shown within a grey border representing the phone's frame.

Registration - Mobile app

3.6.2.3 Change Password

This feature allows users to update their existing password to a new one. It typically involves a user interface where users can input their current username and password along with the new password they wish to set. Changing passwords regularly is a fundamental security practice. It helps users maintain the security of their accounts, reducing the risk of unauthorized access if the

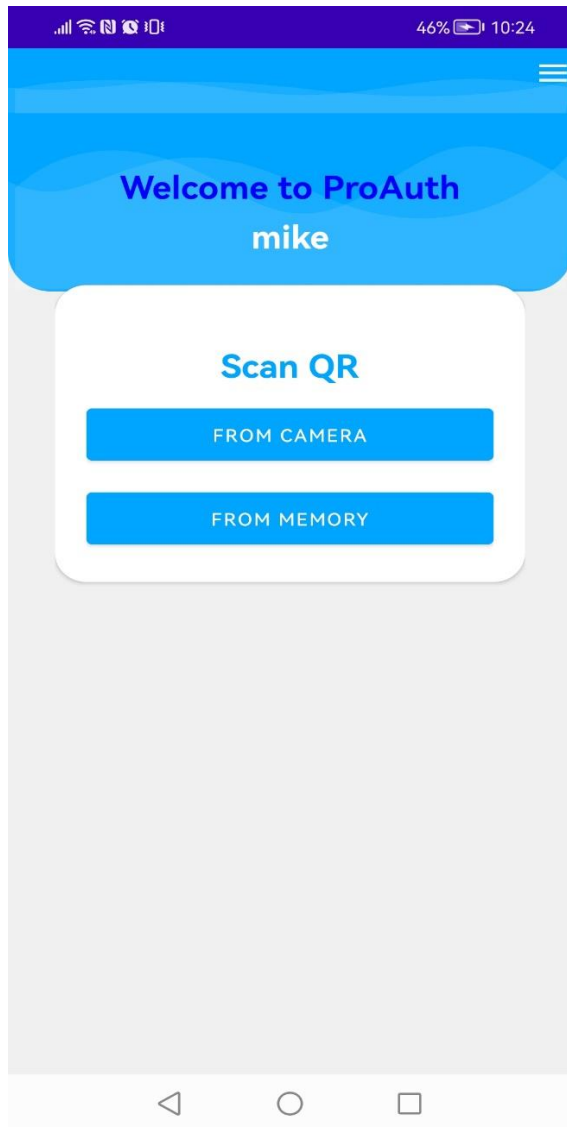
old password is compromised.



Change password - Mobile app

3.6.2.4 Home

Within the home page of the mobile application, we have two main features. Scan the product QR Code “from Camera” and scan “from Memory”.



Home - Mobile app

3.6.2.4.1 Scan from Camera

This option allows to the users to instantly authenticate products by scanning their QR code using the device's camera.

How It Works: Users access the camera feature within the app, allowing them to point their device at a product's QR code. The app captures and processes the QR code data, swiftly verifying the product's authenticity and give back the final result.

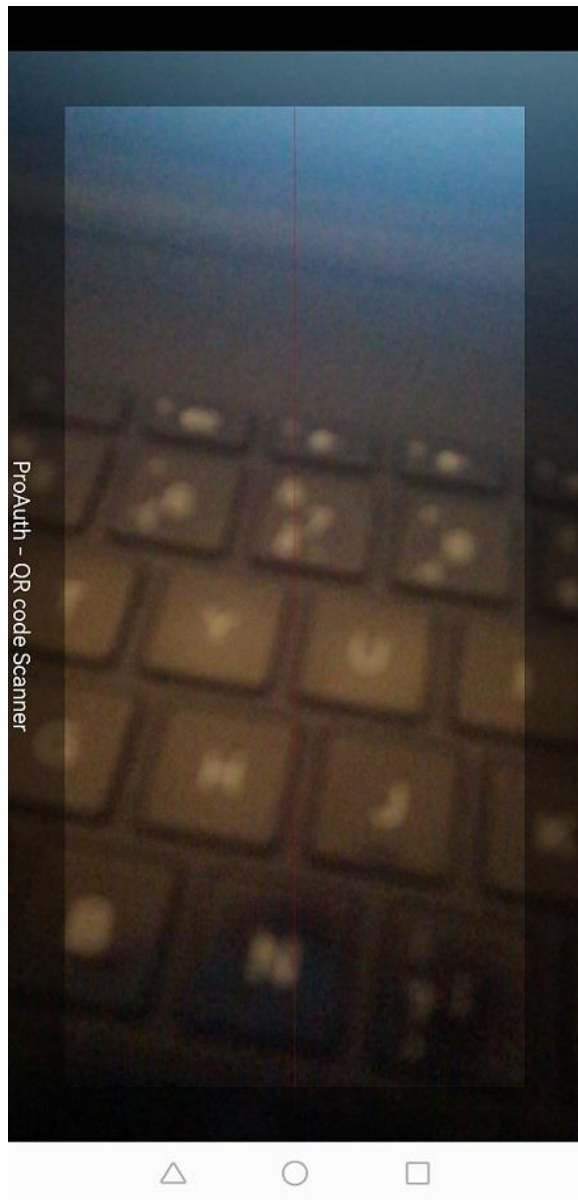
Benefits: This feature offers real-time authentication, providing immediate results while ensuring ease of use through quick and direct access to the camera.

3.6.2.3.2 Scan from Memory

This option allows to the users to authenticate products by selecting QR code images from the device's gallery or saved photos.

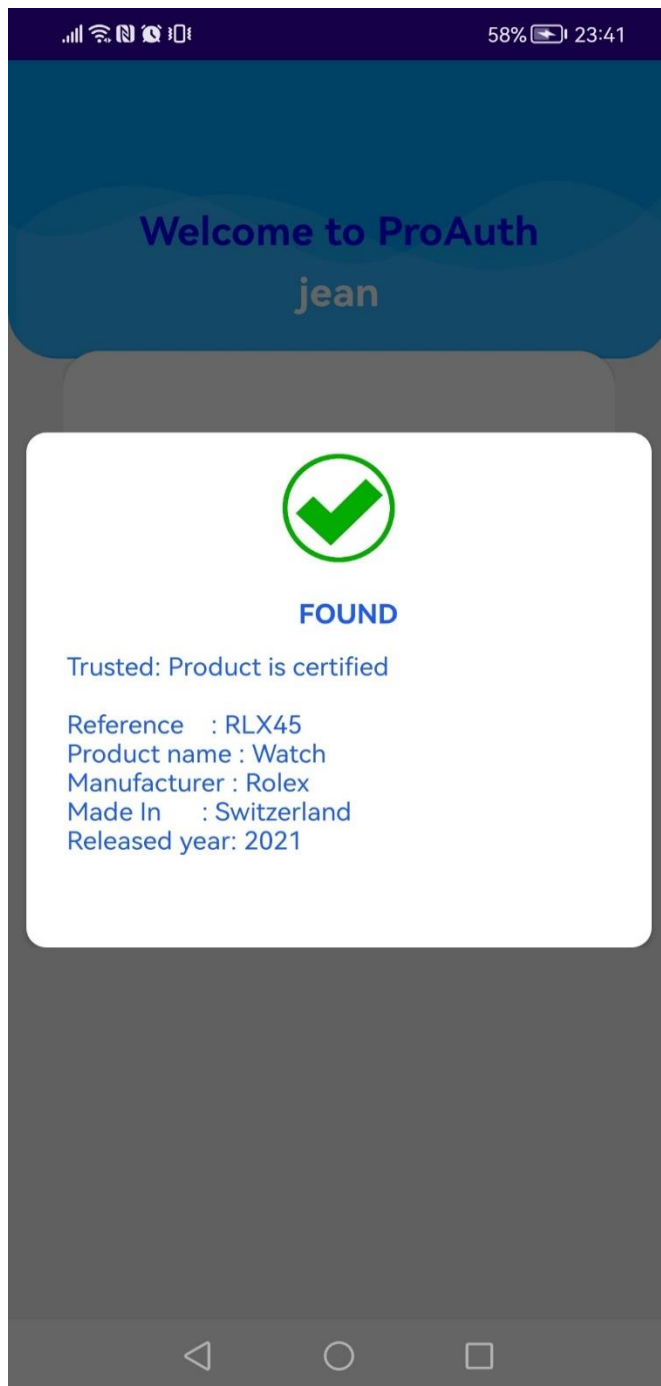
How It Works: Users have the option to select saved QR code images from their device's gallery. The application processes the chosen image, extracting and analyzing the QR code data to verify the product's authenticity and give back the final result.

Benefits: Provides flexibility by allowing users to authenticate products without needing to scan in real-time. Users can select saved images, ensuring convenience and usability.



QR Code scanner - Mobile app

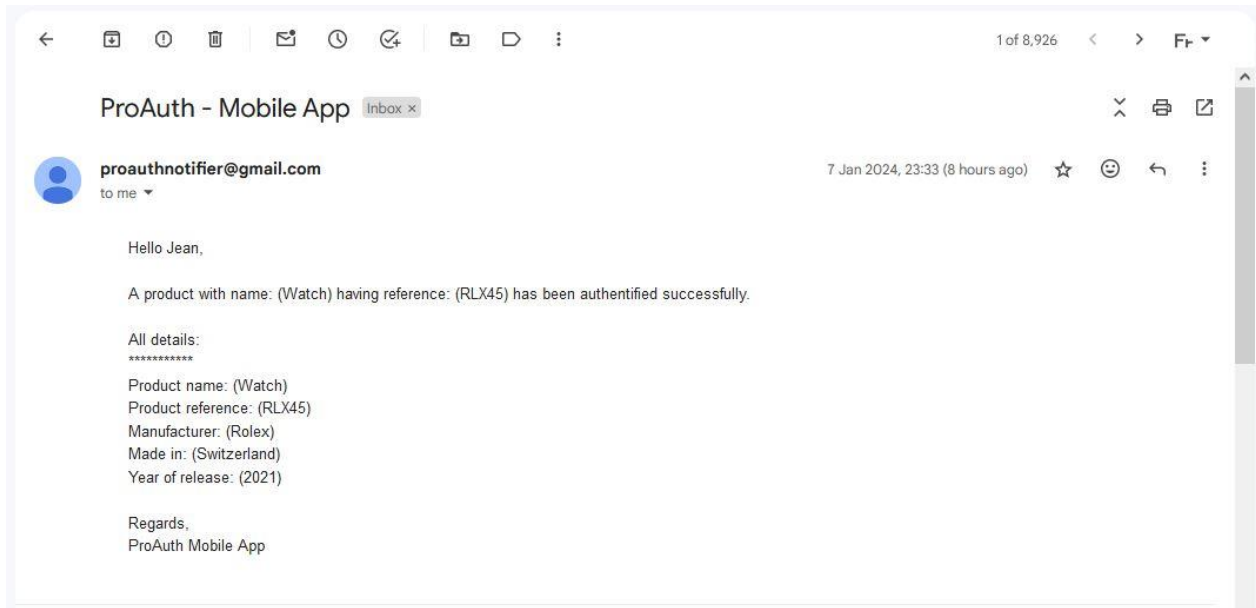
Once a product is authenticated successfully, “FOUND” status is returned to the user with more details related to the product such as the reference, product name, manufacturer, origin country and the year of release.



Result from QR Code scanner – Mobile app

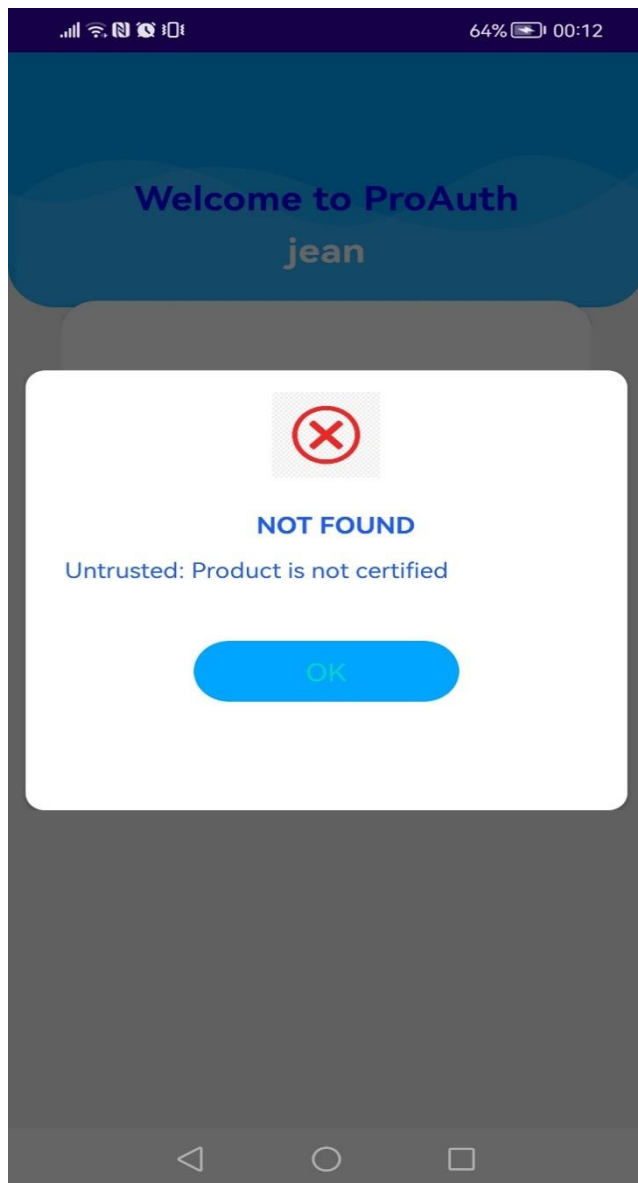
If the notification setting is enabled within the settings of the user in the ongoing session, an automated email is generated and sent to the user's registered email address. The email includes detailed information about the authenticated product. This feature provides users with a documented record of their product authentication activities for future reference or validation

purposes.



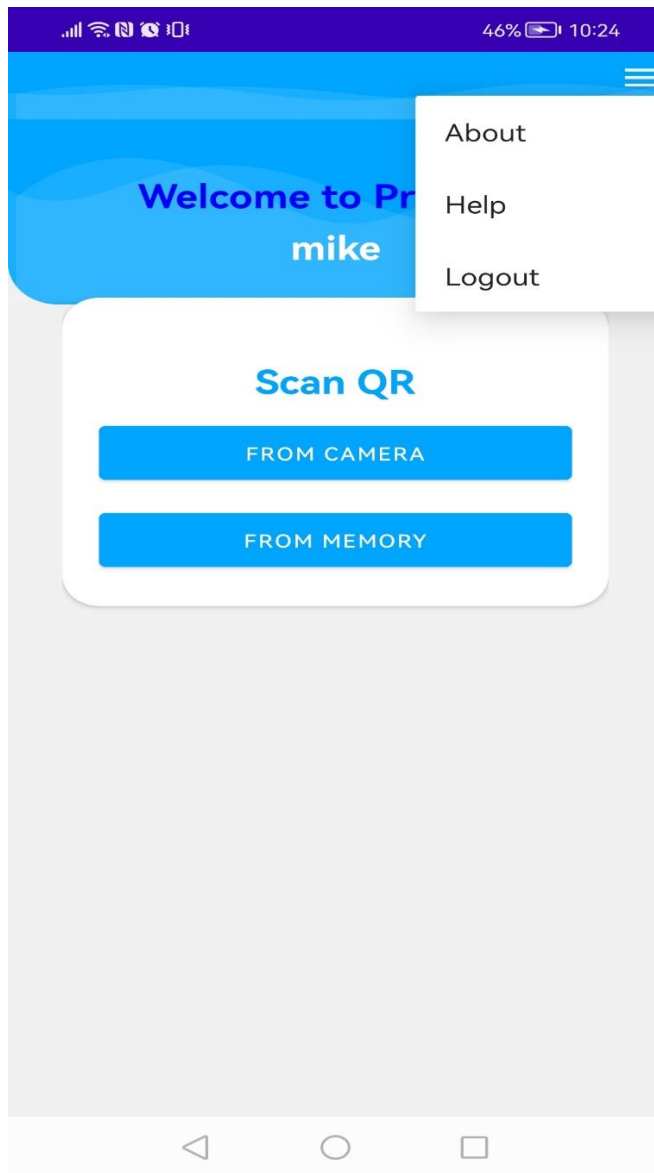
Email notification – Sample

Otherwise, “NOT FOUND” status is returned once a product is not authenticated successfully as in the following image below.



Product not found – Mobile app

3.6.2.5 Menu



In menu section, we have 3 options: About, Help and Logout

About: Describe the main purpose of the application

Help: Describe the features implemented in the application and how they are working

Logout: Allow the user to quit the application in a secure way.

CHAP IV. DISCUSSION

4.1 Benefits of the current system

- The current system provides a reliable and transparent way of verifying the authenticity of products using a web and mobile technologies.
- Blockchain ensures that the product information is immutable and decentralized, preventing tampering and fraud by malicious actors.
- QR code enables a convenient and user-friendly interface for customers to scan the product and access its details from the database.
- The system also sends an email notification to the customer upon scanning, confirming the product verification and providing additional information.
- The system leverages the advantages of both blockchain and database, such as security, scalability, performance, and availability.
- The system can be applied to various domains and industries that require product identification and verification, such as pharmaceuticals, luxury goods, food, etc.

4.2 Limitations

- A main limitation is the user experience of the mobile app, which requires internet connection and QR code scanner to access the product information. If the internet connection is slow or unavailable, or if the QR code is damaged or unreadable, the user may not be able to verify the product authenticity. Moreover, the user may not trust the information provided by the app, especially if they are not familiar with blockchain technology.
- The second limitation of this solution is the scalability of the blockchain network, which depends on the number of nodes, transactions, and consensus mechanism. The more nodes and transactions, the more computational power and storage space are required. The consensus mechanism also affects the speed and security of the network.
- Another limitation is the cost of using the blockchain network, which involves paying gas fees for every transaction. Gas fees are determined by the supply and demand of the network, and can

fluctuate significantly depending on the network congestion and market conditions. Gas fees can be a barrier for small and medium-sized enterprises (SMEs) that want to use the proposed system.

- A fourth limitation is the security of the QR code, which can be tampered with or duplicated by malicious actors. Although the QR code is encrypted and linked to the blockchain, it is still possible to create fake QR codes that point to fake product information. Therefore, additional security measures are needed to prevent QR code fraud.

- A fifth limitation is the legal and regulatory aspects of using blockchain technology for product identification. Different countries may have different laws and regulations regarding blockchain technology, data privacy, consumer protection, and product liability. Therefore, the proposed system may face legal and ethical challenges in different jurisdictions.

4.3 Future implementations

Previously, we discussed the limitations of our proposed system for product identification using blockchain and QR code. We also have in our pipeline some possible future implementations that could enhance the functionality and security of the system.

This system has some limitations and challenges that need to be addressed in future work. Some of these are:

- The user experience and adoption. Although our system aims to provide a convenient and user-friendly way of verifying product authenticity, it still requires some effort from the users to scan the QR code and check the email notification. Moreover, some users may not have access to a smartphone or internet connection. A possible solution is to educate and incentivize the users about the benefits and features of our system, as well as to provide alternative ways of accessing the product information without internet such as USSD and SMS.

- The security of the QR code. Although we use a secure QR code that encrypts the product information and prevents unauthorized access, there is still a risk of fraud if someone copies or modifies the QR code. For example, a counterfeiter could attach a fake QR code to a genuine product, or vice versa. A possible solution is to add a watermark to the QR code that can be detected by the mobile app and verify its integrity. In this case an attempt of duplicating or copying a QR code will make it invalid as it will lose its original pixel quality.

- The scalability of the blockchain network. As more products are registered in the blockchain, the network becomes more congested and the transaction fees increase. This could affect the performance and cost-effectiveness of our system. A possible solution is to use a layer 2 solution, such as Plasma, that can process transactions off-chain and reduce the load on the main chain.
- Adding advanced settings to the mobile app, such as language preference, notifications.
- Allowing users to rate and comment on the products they scan, creating a feedback loop and a social network effect.

CHAP V. CONCLUSION

In conclusion, the system for product authentication using web and mobile application is a novel and promising solution that could benefit both the manufacturer and the customer in terms of authenticity, security, and convenience. However, it also has some challenges and limitations that need to be overcome in future work. The system is a proof-of-concept that demonstrates the feasibility and potential of using blockchain and QR code technologies for product identification purposes so that the customers ensure about the originality and authenticity of a given product before buying. However there's a need of pursuing with some additional features in the future to improve the user experience and the ecosystem itself.

BIBLIOGRAPHY

1. Leon Shklar, Richard Rosen, *Web Application Architecture: Principles, Protocols and Practices*, John Wiley & Sons, Ltd, pp. 1-2
2. Dominic Chell, Tyrone Erasmus, Shaun Colley and Ollie Whitehouse, *The Mobile Application Hacker's Handbook*, WILEY, pp. 2-4
3. Daniel Drescher, *Blockchain Basics: A Non-Technical Introduction in 25 Steps*, Apress, Pp. 16,33-35
4. Imran Bashir, *Mastering Blockchain: Distributed ledger technology, decentralization and smart contracts explained*, Packt, The second edition of the book, pp. 55
5. *Client-server software design pattern*, ALTITUDE ACCELERATOR:
<https://altitudeaccelerator.ca/software-architecture-design-patterns/> (Accessed on 29/10/2023)
6. *Blockchain*, Wikipedia: <https://en.wikipedia.org/wiki/Blockchain> (Accessed on 02/11/2023)
7. *How Does the Blockchain Work?*, Geeksforgeeks: <https://www.geeksforgeeks.org/how-does-the-blockchain-work/> (Accessed on 02/11/2023)
8. *What is blockchain?*, McKinsey & Company: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-blockchain> (Accessed on 05/11/2023)
9. *Blockchain technology: What it is, benefits, and its cross-industry applications*, INSIDER INTELLIGENCE: <https://www.insiderintelligence.com/insights/blockchain-technology-applications-use-cases/> (Accessed on 05/11/2023)
10. *What is a QR code?*, BUSINESS INSIDER: <https://www.businessinsider.com/guides/tech/what-is-a-qr-code?IR=T> (Accessed on 06/11/2023)
11. *What is blockchain technology?*, IBM: <https://www.ibm.com/topics/blockchain> (Accessed on 06/11/2023)
12. *What is Blockchain Technology?*, AWS: <https://aws.amazon.com/what-is/blockchain/?aws-products-all.sort-by=item.additionalFields.productNameLowercase&aws-products-all.sort-order=asc> (Accessed on 08/11/2023)

13. *Android Applications and Their Categories*, Geeksforgeeks: <https://www.geeksforgeeks.org/android-applications-and-their-categories/> (Accessed on 09/11/2023)
14. *CSS*, Wikipedia: <https://en.wikipedia.org/wiki/CSS> (Accessed on 10/11/2023)
15. *What is JavaScript?* Mozilla: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript (Accessed on 10/11/2023)
16. *What Is Node.JS and What Is It Used for?*, How-to Geek: <https://www.howtogeek.com/devops/what-is-node-js-and-what-is-it-used-for/> (Accessed on 11/11/2023)
17. *Express JS Tutorial*, Simplilearn: <https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-express-js> (Accessed on 11/11/2023)
18. *What is Ethereum?*, Euthereum: <https://ethereum.org/en/> (Accessed on 11/11/2023)
19. *Ethereum For Beginners*, Blockchain: <https://www.blockchain.com/learning-portal/ether-basics> (Accessed on 12/11/2023)
20. *Ganache Blockchain: All you need to Know*, Blockchain Council: <https://www.blockchain-council.org/blockchain/ganache-blockchain-all-you-need-to-know/> (Accessed on 13/11/2023)
21. *MetaMask*, Wikipedia: <https://en.wikipedia.org/wiki/MetaMask> (Accessed on 14/11/2023)
22. *What is Truffle in Blockchain?*, Shardeum: <https://shardeum.org/blog/truffle/> (Accessed on 14/11/2023)
23. *What Is MongoDB?*, MongoDB: <https://www.mongodb.com/what-is-mongodb> (Accessed on 17/11/2023)
24. *What is MongoDB - Working and Features*, Geeksforgeeks: <https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/> (Accessed on 18/11/2023)
25. *What is MongoDB?*, GURU99: <https://www.guru99.com/what-is-mongodb.html> (Accessed on 19/11/2023)
26. *Postman Tutorial*, Javatpoint: <https://www.javatpoint.com/postman> (Accessed on 01/12/2023)

APPENDIX

Source code: Login Restful Service for mobile application

```
234 // Retrieve a user by username and password - Mobile login
235 app.post('/login-mob', async (req, res) => {
236   try {
237     const { username, password } = req.body;
238     const userMobile = await UserMobile.findOne({ username });
239
240     if (!userMobile) {
241       return res.status(401).send('User not found');
242     }
243
244     const isMatch = await bcrypt.compare(password, userMobile.password);
245
246     if (!isMatch) {
247       return res.status(401).send('Invalid password');
248     }
249     //Capture logged in
250     const loggedUser = new VisitLog({
251       ...req.body,
252       loggedInAt: new Date()
253     });
254     await loggedUser.save();
255
256     res.status(200).send('Login successful');
257   } catch (error) {
258     res.status(500).send('Error logging in');
259   }
260 });
261
```

Source code: Change Password Model for mobile application

```
3 public class UserChangePassword {
4
5     3 usages
6     private String username;
7     3 usages
8     private String password;
9     3 usages
10    private String newPassword;
11
12    1 usage
13    public UserChangePassword(String username, String password, String newPassword) {
14        this.username = username;
15        this.password = password;
16        this.newPassword = newPassword;
17    }
18
19    public String getUsername() { return username; }
20
21    public void setUsername(String username) { this.username = username; }
22
23    public String getPassword() { return password; }
24
25    public void setPassword(String password) { this.password = password; }
26
27    public String getNewPassword() { return newPassword; }
28
29    public void setNewPassword(String newPassword) { this.newPassword = newPassword; }
30
31    }
32
33    }
```

Source code: Registration Activity for mobile application

```
159     private void registrationApiCaller(final String firstnameVal,|
160                                     final String lastnameVal,
161                                     final String usernameVal,
162                                     final String passwordVal,
163                                     final String addressVal,
164                                     final String emailVal,
165                                     final String phoneVal,
166                                     final String countryVal,
167                                     final boolean emailNotificationVal) {
168
169         // Displaying our progress bar.
170         loadingPB.setVisibility(View.VISIBLE);
171
172         final UserApiService userApiService = APIClient.getClient().create(UserApiService.class);
173
174         final UserRegistration userRegistration = new UserRegistration(firstnameVal,
175                                 lastnameVal,
176                                 usernameVal,
177                                 passwordVal,
178                                 addressVal,
179                                 emailVal,
180                                 phoneVal,
181                                 countryVal,
182                                 emailNotificationVal);
183
184         Call<String> call = userApiService.registerNewUser(userRegistration);
185         call.enqueue(new Callback<String>() {...});
209     }
```

Source code: Build blockchain smart contract for product

```
3  contract ProductDetails {
4      //State variable - State of the contract inside the blockchain
5      uint public productCount = 0;
6
7      struct ProductEntity{
8          uint id;
9          /*string name;
10         string referenceId;
11         string manufacturer;
12         string madeIn;
13         string description;
14         string category;*/
15
16         string productDetails; //name , referenceId, manufacturer, madeIn, description, category,
17         uint256 price;
18         uint256 warranty;
19         uint16 yearOfRelease;
20         //string createdBy;
21         uint256 createdAt;
22     }
23
24     mapping(uint => ProductEntity) public products;
25
26     event ProductCreated(
27         uint id,
28         string productDetails,
29         uint256 price,
30         uint256 warranty,
31         uint16 yearOfRelease,
32         uint256 createdAt
33     );
34 }
```

Source code: Display products from the blockchain

```
269 searchProduct: async () => {
270   //Load the total task count from the blockchain
271   var prodId = $('#prodId').val()
272   //prodId = parseInt(prodId)
273   //const productCountCp = await App.productListCp.productCountCp()
274   var productCount = await App.productDetails.productCount();
275   const $searchProductDiv = $('#tableSearchProductDiv')
276   const $prodSearchTemplateNotFound = $('#prodSearchTemplateNotFound')
277
278   const tableBody = document.querySelector('#tableProductSearch tbody');
279   tableBody.innerHTML = '';
280
281   const $qrcodeTemplate = $('#qrcode')
282   $qrcodeTemplate.empty()
283   console.log("Search="+ prodId)
284   console.log("Count="+ productCount)
285   //Render out each task with a new task template
286   if(parseInt(prodId) > 0 && parseInt(prodId) <= parseInt(productCount)){
287     console.log("Inside="+ productCount)
288     //Fetch the task data from the Blockchain
289     const product = await App.productDetails.products(prodId)
290     const productId = product[0].toNumber()
291     //name , referenceId, manufacturer, madeIn, description, category, createdBy
292     const productDetails = product[1].split("|");
293     const productName = productDetails[0];
294     const productReferenceId = productDetails[1];
295     const productManufacturer = productDetails[2];
296     const productMadeIn = productDetails[3];
297     const productDescription = productDetails[4];
298     const productCategory = productDetails[5];
299     const productCreatedBy = productDetails[6];
300
301     const productPrice = product[2];
302     const productWarranty = product[3];
303     const productYearOfRelease = product[4];
304     const productCreatedAt = product[5];
305   }
```

Source code: Login activity for mobile application

```
23 <com.google.android.material.textfield.TextInputLayout
24     android:id="@+id/textInputEmail"
25     style="@style/parent"
26     android:layout_marginTop="20dp">
27     <EditText
28         android:id="@+id/editTextUsername"
29         style="@style/modifiedEditText"
30         android:inputType="text|textUri"
31         android:maxLines="1"
32         android:hint="Username"/>
33 </com.google.android.material.textfield.TextInputLayout>
34
35 <com.google.android.material.textfield.TextInputLayout
36     android:id="@+id/textInputPassword"
37     style="@style/parent"
38     android:layout_marginTop="20dp">
39     <EditText
40         android:id="@+id/editTextPassword"
41         style="@style/modifiedEditText"
42         android:hint="Password"
43         android:maxLines="1"
44         android:inputType="textPassword"/>
45 </com.google.android.material.textfield.TextInputLayout>
46
```