



---

# POZNAN UNIVERSITY OF TECHNOLOGY

---

FACULTY OF COMPUTING AND TELECOMMUNICATION  
Institute of Computing Science

Blockchain technology

## BLOCKCHAIN PROJECT - IMPLEMENTATION

inż. Michał Kaczkowski, 144259  
inż. Wojciech Kaczmarek, 144250

Tutor  
mgr inż. Jakub Hamerliński

POZNAŃ 2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>User Interface</b>	<b>2</b>
<b>3</b>	<b>Security mechanism</b>	<b>4</b>
<b>4</b>	<b>Scalability</b>	<b>5</b>

# Chapter 1

## Introduction

In our project we've implemented simple blockchain with transaction system. Blockchain is implemented in python programming language as a class which consists of two main functions first used for creating a new block and a second one creating new transaction. It also has few more functions such as block hashing, checking if the chain and proof is valid. Function reading and writing transaction list to a file were also implemented. Transactions are store in json file and are loaded to the program after user provides user and password. User credentials are hashed and store in a txt file. After login we can see few options first of which adds a new block (after that new transactions are visible in database. Second option is new transaction where we can send money to other people. The sender is always the person logged in at the moment. Third option shows us the whole blockchain and saves its state to json file. Forth option checks if the blockchain is valid and fifth one logs us out.

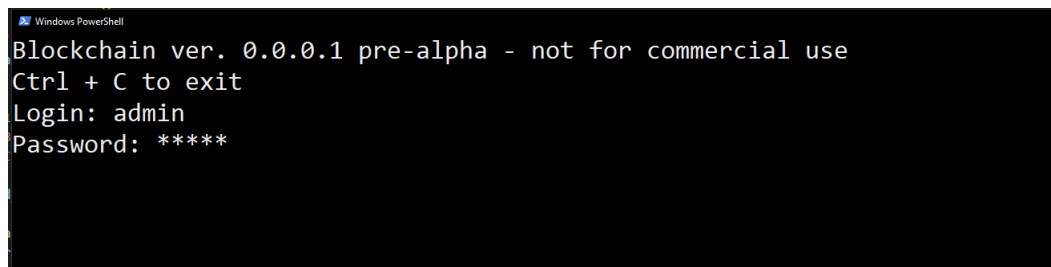
## Chapter 2

# User Interface

As user interface, due to fact, this is pre-alpha version, we use CLI to interact with our program. As we mentioned in introduction, there is logging panel and sort of menu to interact with blockchain. We have 5 options to choose after logging:

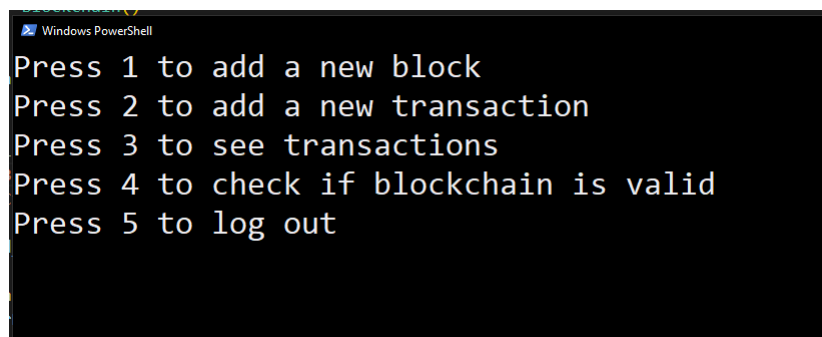
- Add new block,
- Add new transaction,
- Print whole blockchain,
- Check, if blockchain is valid,
- Log out from app.

Below, we show some screenshots from the app.



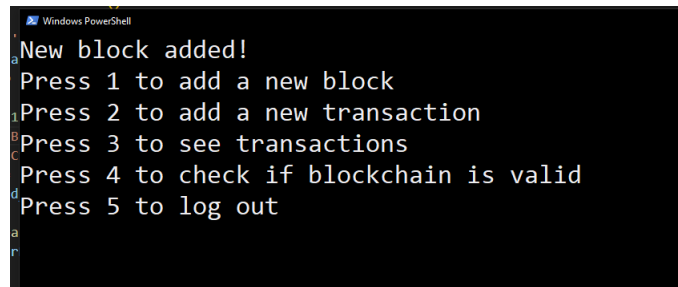
```
Windows PowerShell
Blockchain ver. 0.0.0.1 pre-alpha - not for commercial use
Ctrl + C to exit
Login: admin
Password: ****
```

FIGURE 2.1: Logging screen



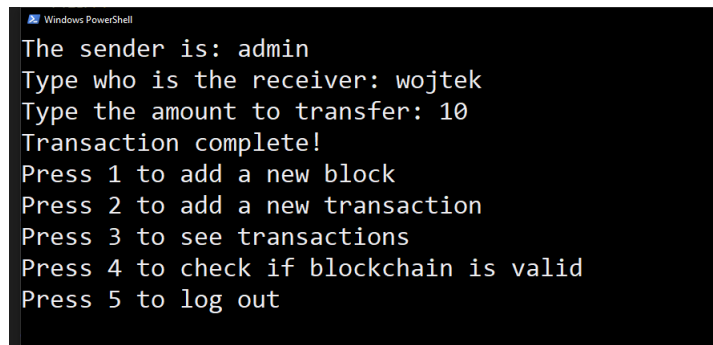
```
Windows PowerShell
Press 1 to add a new block
Press 2 to add a new transaction
Press 3 to see transactions
Press 4 to check if blockchain is valid
Press 5 to log out
```

FIGURE 2.2: Main menu



```
Windows PowerShell
New block added!
Press 1 to add a new block
Press 2 to add a new transaction
Press 3 to see transactions
Press 4 to check if blockchain is valid
Press 5 to log out
```

FIGURE 2.3: First function - add new block



```
Windows PowerShell
The sender is: admin
Type who is the receiver: wojtek
Type the amount to transfer: 10
Transaction complete!
Press 1 to add a new block
Press 2 to add a new transaction
Press 3 to see transactions
Press 4 to check if blockchain is valid
Press 5 to log out
```

FIGURE 2.4: Second function - add new transaction



```
-----
{
  "index": 14,
  "timestamp": 1685994953.720696,
  "transactions": [
    {
      "sender": "wojtek",
      "recipient": "kuba",
      "amount": "333"
    }
  ],
  "proof": 72975,
  "previous_hash": "dd16b6de734cf4dad7ef7d771fedce0b61646919d4bba79fbaf401b674b0a068"
}
-----
Blockchain exported to file: sample.json
Press 1 to add a new block
Press 2 to add a new transaction
Press 3 to see transactions
Press 4 to check if blockchain is valid
Press 5 to log out
```

FIGURE 2.5: Third function - show whole blockchain

Option number 4 just return us value of True or Flase if blockchain is valid. Option number 5 log out current user and bring program to logging page.

## Chapter 3

# Security mechanism

Main security mechanisms are:

- Hashing

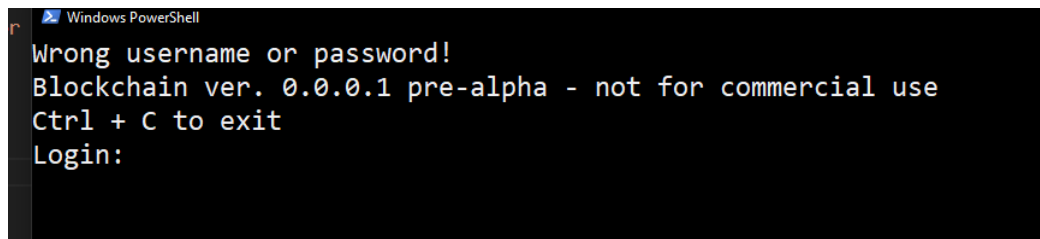
Hashing are used in two places:

- Users passwords are stored as **SHA** hash
- Chains are connected by hash, calculated from previous hash

- proof-of-work

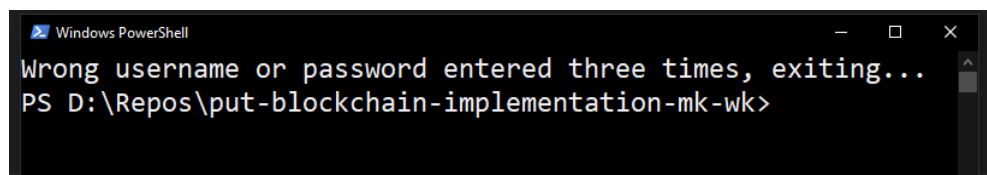
Proof of work is random number added to blockchain. Due to fact, that the hash is calculated from blocks, any change in this value will drastically change hash.

We have also implemented protection against brute forcing app, after 3 failed attempts to login, app will closed itself.

A screenshot of a Windows PowerShell terminal window. The text displayed is: "Wrong username or password!", "Blockchain ver. 0.0.0.1 pre-alpha - not for commercial use", "Ctrl + C to exit", and "Login:". The prompt is a red character.

```
Wrong username or password!
Blockchain ver. 0.0.0.1 pre-alpha - not for commercial use
Ctrl + C to exit
Login:
```

FIGURE 3.1: Wrong password or username warning

A screenshot of a Windows PowerShell terminal window. The text displayed is: "Wrong username or password entered three times, exiting..." and "PS D:\Repos\put-blockchain-implementation-mk-wk>". The window title bar shows "Windows PowerShell" and standard window controls.

```
Wrong username or password entered three times, exiting...
PS D:\Repos\put-blockchain-implementation-mk-wk>
```

FIGURE 3.2: Failed login log

## Chapter 4

# Scalability

Storing the blockchain in a .json file can facilitate its scalability for several reasons. There are several benefits to using the .json (JavaScript Object Notation) format in the context of blockchain development and expansion.

Firstly, the .json format is widely used in programming and inter-system communication. It is a human-readable text-based format, making it easy to create and analyze data. Storing the blockchain in a .json file enables straightforward data manipulation using various programming languages and tools, which is crucial for system development and scalability.

Secondly, the .json format is highly adaptable to changing needs and blockchain extensions. With its flexible structure, it is simple to add new fields and attributes to the .json file, allowing the blockchain to be customized to evolving requirements and functionalities. This is particularly important in the development of blockchain applications, where continuous adaptation to a changing environment is necessary.

Another aspect is the compatibility of the .json format with databases and analytical tools. Many developers and analysts have experience working with databases that support the .json format, making it easier to integrate the blockchain with existing systems. This enables easy processing, analysis, and utilization of data stored in the .json file for generating reports, visualization, and decision-making.

Additionally, the .json format is platform-independent, meaning it can be used across different operating systems and devices. This is essential for blockchain scalability, where different network nodes may operate on various platforms. Storing blockchain data in the .json format ensures a uniform structure that can be read and processed regardless of the platform, facilitating interoperability and communication between nodes.

In summary, storing the blockchain in a .json file can enhance its scalability by providing ease of data manipulation, adaptability to changing needs, compatibility with databases and analytical tools, and platform independence. It is a solution that allows for flexibility and facilitates the expansion of blockchain technology.