Blockchain Based Vaccine Verification System

07.09.2023

Elif Rüveyha Özmen, Alessandro Migotto

Università degli studi di Trento
Blockchain [146157]
Luigi Telesca, Mahfuzul Islam

INDEX

Abstract and introduction

In our ever-evolving world where health crisis and concerns about data privacy have been increasing, there is a need for an efficient, secure system to verify individuals' vaccination records. Current vaccine record verification systems are often cumbersome, require the sharing of personal data, and lack the integrity of the data. Furthermore, vaccine record storage varies between countries and even among different health centers. In some cases, keeping records is the responsibility of individuals, potentially leading to record loss.

To better understand this concept we can use a simpler real-world scenario, for example, the Covid pandemic period in Italy. During that period, in order to access public places such as bars or hospitals, a “Green Pass” was required. This special pass was granted to people only after taking the appropriate number of Covid vaccines mandated by the Government. Once you had shown this pass to the relevant authority and scanned your QR code, they would access your personal data and cross-check with your identity card to see if you were really who you claimed to be. Additionally, all this data was stored somewhere, typically in a database. However, you as an individual, had limited knowledge and control over how this data is stored and processed and there was no guarantee of protection against potential data breaches.

This project addresses these issues by leveraging blockchain's capabilities to develop a solution that securely stores the data and accurately verifies vaccination records without evidently keeping information about them and therefore enforcing their privacy against public and private entities entitled to check such eligibility in order to provide access to their structures.

The system we developed provides a user-friendly interface with which everyone can demonstrate if they have a particular vaccine, as well as their eligibility to access places, anytime and everywhere, without the need of carrying extra documentation. Moreover, adding vaccines to the system bound to a specific user is totally secure and can be performed and monitored only by competent authorities, such as doctors or similar. The blockchain model ensures transparency and privacy and availability of the data.

The Blockchain-Based Vaccine Verification System (BBVV) is an innovative way to utilize blockchain technology through a decentralized application to address the challenges associated with vaccine verification and record-keeping. This comprehensive report provides an in-depth analysis of the DApp's objectives, technology stack, system architecture, and key features.

Amidst the global significance of vaccination in public health, the need for secure, tamper-proof, and easily accessible vaccination records has become paramount. The

DApp aims to revolutionize vaccine verification by leveraging blockchain technology and providing a user-friendly interface for individuals, healthcare providers, and government agencies.
The primary objectives of the DApp are to create a secure and immutable repository of vaccination records, enable easy verification and access to vaccination status, and facilitate seamless communication between stakeholders in the healthcare ecosystem.

Technology stack & background

Blockchain technology has garnered significant attention for its potential to revolutionize various industries beyond finance. The key features that make blockchain attractive for applications like the Vaccine Verification System are its decentralized nature, immutability, and transparency.
A blockchain is a distributed and decentralized ledger that records all transactions in a secure and immutable manner. Each transaction, also known as a block, is cryptographically linked to the previous one, forming a chain of blocks. This ensures that any attempt to alter data in a block would require modifying all subsequent blocks, making it practically impossible to tamper with the data without detection.
The concept of vaccine verification systems is not entirely new, as numerous initiatives and projects have attempted to create centralized databases for vaccine records. However, these systems often face challenges related to data privacy, security breaches, and interoperability. The inherent transparency and security of blockchain technology offer a promising solution to these issues.
This project utilizes the Ethereum blockchain, Solidity smart contracts, React frontend, and Truffle development framework to create a decentralized application for managing user authorization and vaccine records.

System Architecture

This application adopts a distributed and decentralized system architecture to ensure robustness and resilience. The architecture, besides frontend and backend, obviously relies on the Ethereum blockchain.

Frontend

The frontend of the DApp is developed using React, a popular JavaScript library for building user interfaces. React's component-based architecture and virtual DOM allow developers to create interactive and responsive user interfaces efficiently. The user interface is designed to be visually appealing, intuitive, but most importantly easy to use.

The frontend component is the user-facing part of the DApp, where users interact with the system through a web-based interface. It enables users to perform various actions, such as verifying vaccination status or checking the eligibility for accessing certain places. Web3.js is utilized on the frontend to connect with the Ethereum blockchain. It enables the frontend to send transactions, and retrieve data, providing users with immediate feedback on the transaction status.

Backend

The backend comprises three essential smart contracts developed in Solidity: System Manager, Vaccine Registry, and Eligibility Checker. These smart contracts define the rules and logic of the DApp. The System Manager contract is the core one, from which the others depend, and it manages the administrative operations; the Vaccine Registry contract stores and manages vaccination records; the Eligibility Checker contract verifies eligibility criteria for gaining access to specific entities with a certain set of required vaccines. These smart contracts were developed and tested using Remix IDE. After verifying the accuracy of the functions used in the smart contracts through Remix IDE, the developed smart contract was tested on the Ethereum blockchain simulation environment Ganache provided by Truffle.

Smart Contracts

The DApp implements three crucial smart contracts to manage various functionalities:

- System Manager: The System Manager contract is designed to manage the distribution of authority and vaccine records on the blockchain. Its main functionalities are vaccine code registration, entity registration, and user authorization. The contract has several components and functionalities:
    - *constructor*: The constructor initializes the admin variable with the address of the deployer.

- *admin*: The address of the contract deployer that can perform several actions that only the admin is entitled to.
- *Entity*: A struct that contains:
  - *entityAddress*: Ethereum address of the entity.
  - *entityName*: String type name of the entity.
  - *vaccineRequirementList*: An array of strings representing vaccine codes required by the entity.
- *OnlyAdmin()*: A custom modifier that allows certain functions to be executed only by the admin. If the sender has a different address than the admin's, the modifier throws an error.
- *authorizedUserList*: A mapping that associates addresses with a boolean value indicating whether the addresses are authorized users.
- *vaccineList*: A mapping that links the string type vaccine codes to boolean values.
- *addressToEntity*: A mapping that associates addresses with an *Entity* struct.
- *entityList*: A mapping that associates the entity indices with uint256 type to their corresponding entity addresses.
- *Events*: The contract emits several events to notify external systems about changes.
  - *EntityAdded*: Emitted when a new entity is registered.
  - *AuthorizedUserAdded*: Emitted when an authorized user is added to the system.
  - *VaccineAdded*: Emitted when a new vaccine code is added to the system.
  - *RequirementAdded*: Emitted when a new vaccine requirement is associated with an entity.
- Functions:
  - *addEntity*: Allows the admin to register new entities to the system by providing an address and a name.
  - *addAuthorizedUser*: Allows the admin to authorize users by adding their addresses.
  - *addVaccineToSystem*: Allows the admin to add new vaccine codes to the system.
  - *addRequirement*: Allows the admin to add vaccine requirements to entities.
  - *getVaccineRequirementList*: An external view type function that retrieves the list of required vaccines for a specific entity.
  - *entityExists*: An external view type function that checks if an entity exists based on its index.

- Vaccine Registry: The Vaccine Registry is responsible for storing vaccination to keep track of recipients' vaccine records. The contract interacts with the SystemManager contract to ensure authorized access and validation of vaccine codes. The contract has several components and functionalities:
    - *sm*: A public variable that holds an instance of the *SystemManager* contract.
    - *constructor*: The constructor initializes variable sm with a provided deployed SystemManager contract address.
    - *recipientVaccineStatus*: A mapping that associates a hash of the recipient's information and vaccine code with a boolean value, indicating vaccination status.
    - *Recipient*: A struct that contains:
        - *id*: String type identifier of the recipient.
        - *countryCode*: String type code of the country of the recipient.
    - *OnlyAuthorized()*: A modifier that ensures only authorized users, as determined by the *SystemManager* contract, can use certain functions.
    - Events:
        - *vaccineAddedToRecipient*: Emitted when a hashed vaccine information is added to the system.
    - Functions:
        - *addVaccineToRecipient*: Allows authorized users to add hashed vaccine record information to the system.
        - *hashRecipientVaccineInfo*: A private function that computes and returns the hash of the recipient and the vaccine information.
        - *viewAuthorizedUsers*: A view type of function to check whether an address is authorized by the sm or not.
        - *isVaccinated*: A view type of function to check whether a recipient is vaccinated for a specific vaccine code.
- Eligibility Checker: The Eligibility Checker contract is designed to determine the eligibility of a recipient for different entities based on their vaccine requirements and the vaccine of the recipients. It ensures that only eligible individuals can access certain entities. The contract has several components and functionalities:
    - *sm*: A public variable that holds an instance of the SystemManager contract.
    - *vr*: A public variable that holds an instance of the VaccineRegistry contract.
    - *constructor*: The constructor initializes variables *sm* and *vr* with the deployed contract addresses of SystemManager and VaccineRegistry.
    - Functions:

        - *isEligible*: Determines whether a recipient is eligible for accessing an entity according to the requirements of an entity and the vaccination of the recipient. It takes *entityIndex* and *Recipient* struct as inputs and returns a boolean value indicating eligibility. It iterates through the vaccine requirement list of the specified entity, checking if the recipient is vaccinated for each required vaccine using the *isVaccinated* function in *VaccineRegistry* contract.

Ethereum Blockchain

The Ethereum blockchain serves as the decentralized ledger that stores all vaccination records in a tamper-proof and immutable manner. Each vaccine transaction is recorded as a block on the blockchain, forming an unbroken chain of data. The decentralized nature of the blockchain ensures that no single entity has control over the data, mitigating the risk of data manipulation or unauthorized access.

Types of account

In this system there will be four types of user accounts with different roles and capabilities:

- System Administrator: in charge of granting the status of “Authorized User” to any node of the blockchain. Moreover, they can add new vaccines and entities to the system, and create the list of vaccines required for each entity. In a real world scenario, this account should be impersonated by the central statal authority, for example, the Ministry of Health, in charge of choosing the recognized vaccines and delegating subordinated authorities.
  Only one node can be admin, and this happens when the blockchain is initialized for the first time and the smart contracts are deployed. Afterward, any of the previously stated operations are therefore safeguarded from unauthorized executions.
- Authorized User: authorized to register vaccines to the recipients by entering the vaccine code, and the unique user identifier of the recipient. Additionally, this account has all the capabilities of a normal user. This account can be granted, for example, to doctors or to whoever should be in charge of registering the vaccine information.
- Entities: an entity can be any of the public or private facilities for which a citizen is required to have any set of mandatory vaccines in order to access. Those entities

can be for example schools, hospitals, public places, institutional offices, etc. Each entity comes with a modifiable set of required vaccines in order to access.

- Normal users: this account has the faculty of checking if an individual has the vaccine that they claim to have, by inserting a vaccine code and unique recipient identifier. For the identification of the recipients, there can be two different approaches: one (currently used) is to leave the legislative statal authority to assign to each user a unique identifier (e.g. the identity card number or similar) and communicate it through official channels (in this case therefore a normal user is NOT a node of the blockchain). This approach is more simple and scalable than the second and also easier for final users, who just need to hold their unique code. The second approach is much harder to achieve but leads to a more ordered and scalable system, that allows each user to be a node of the Ethereum blockchain and therefore possess an address to uniquely identify them. Furthermore, anyone who is in charge of checking the eligibility of a citizen to enter public structures can also be classified as a normal user.

Features of the DApp

The BBVV System encompasses a comprehensive set of features designed to address the specific needs of various stakeholders. These features include:

- No need for user registration:

The DApp allows users to utilize the system without the need of registering or creating an account. All they need to do is just to keep their Ethereum wallets(with the browser extension MetaMask) to interact with the DApp.

- Vaccine Verification and Issuance:

Users can verify their vaccination status only by submitting their user identifier, country code and the vaccine code as stated. Any other personal data are not required. Healthcare providers and authorized entities can issue digital vaccination certificates on the blockchain.

- Data Privacy and Security:

The DApp employs encryption and private key management techniques to safeguard user data and maintain data privacy. Registering vaccination records is strictly controlled, ensuring only authorized users can. Additionally, all the personal data needed for vaccine registration is cryptographically hashed before the registration, which ensures that no personal data is evidently stored in blockchain.

- Transaction History:

Any node in the network can see and verify any transaction, making the system transparent. The transaction history is transparent and verifiable, providing authorities with a comprehensive overview of the transactions.

- User Interface:

The user interface of the DApp is intuitive, visually appealing, and user-friendly. It features easy navigation and seamless interactions, enhancing the overall user experience.

Implementation & Development Details

The successful implementation of the BBVV System involved planning, development and testing. The development process followed an agile methodology, enabling iterative improvements, meanwhile added and documented in the present report. As for the workload of the project, it has been splitted evenly between the participants, as also most of the project parts have been done while physically together in presence.

- Smart Contract Development (Solidity):

The smart contract development process begins with defining the data structures and functions required for vaccine verification and issuance. It includes mechanisms for data validation, access control, and event logging. Then the smart contracts are thoroughly tested to ensure security and efficiency.

- Frontend Development (React):

Frontend development involves creating the user interface and designing interactive components. The user interface is designed to be responsive, visually appealing, and user-friendly. React components are used to manage the state and facilitate dynamic updates while the Bootstrap framework provides a more visually appealing interface with responsive layouts and pre-styled components.

- Integration of Backend and Frontend:

Web3.js serves as the crucial bridge enabling the integration of the backend (smart contracts) and frontend in our dApp. This library facilitates direct interaction between the user interface and the blockchain's decentralized operations, streamlining data retrieval and transaction processing.

- Testing and Quality Assurance Procedures:

Comprehensive testing procedures, including unit testing on smart contracts' functions, have been conducted to identify and resolve any potential issues or bugs.

User Guide & Instructions

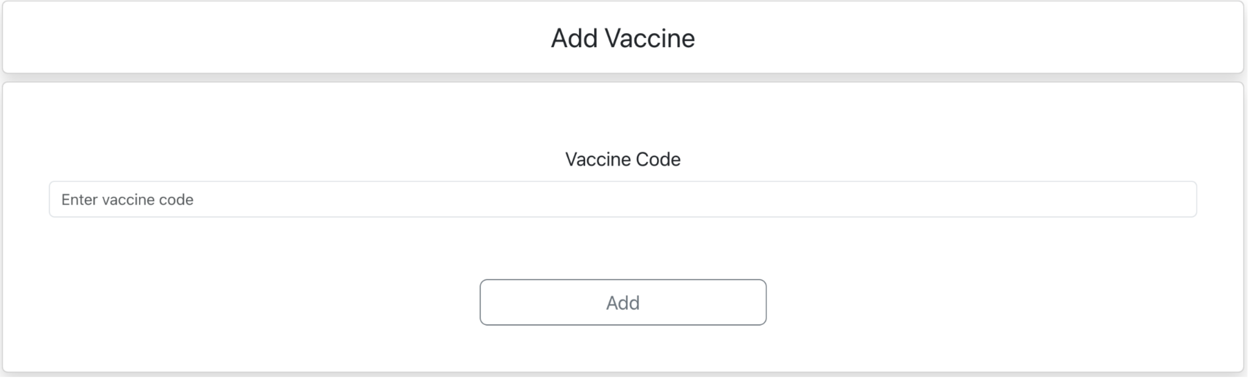
The preferred method for users to access the application interface is through the Metamask program. This program operates as a browser extension, enabling users to log into the application by linking their Ethereum wallets and private keys with Metamask. Within the application, Metamask carries out user verification using digital signatures. Each transaction is confirmed via Metamask, as users load their Ethereum wallets onto Metamask with their private keys. These private keys are used to sign transactions, ensuring user identity verification.
The smart contract is developed with Solidity and then incorporated into the Ethereum blockchain platform. Upon loading the smart contract, the constructor section within the contract is executed. This step defines the admin user, who introduces user privileges to the blockchain network on a one-time basis for application usage. The necessary attributes for the admin user to gain access to the application are specified within the constructor segment.
The user guide provides step-by-step instructions on how to use the BBVV System effectively. It covers every aspect of the DApp, from registration to vaccine verification. Note that the content of the frontend page is dynamic, and changes according to which account is currently logged in the Metamask wallet. In the following, a list of all the action that can be performed by every type of account:

- Actions that can be performed only by system administrator:

In order to perform these kinds of actions, you need to be logged in in your Metamask wallet with administrator privileges, i.e. with the account that deployed first the contracts to the blockchain. These actions include:

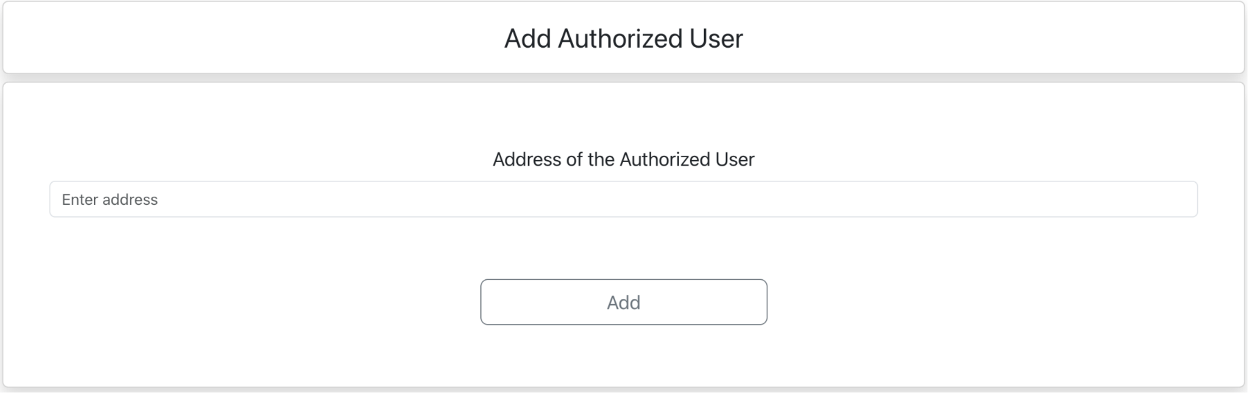
    - *Registration of vaccine (adding a new vaccine to the system):*

Scroll down until the “Add Vaccine” section and then insert the vaccine code. The standardization of vaccine code names is nothing of our concern, but it would be decided and agreed by the state authority.

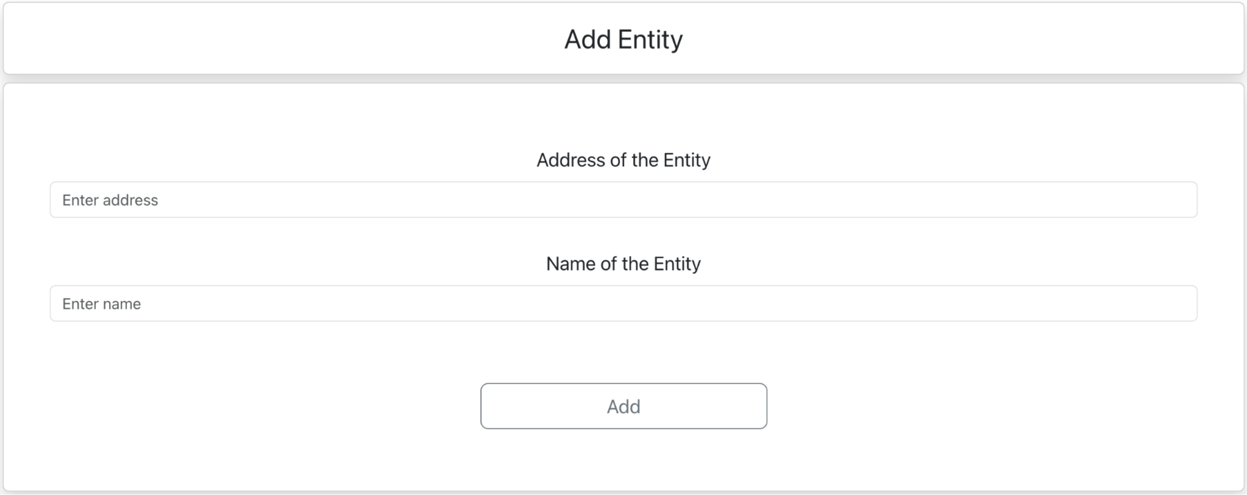


- *Granting an account superior privileges:*

Administrator can decide to make an account become an *authorized user*: scroll down to the “Add authorized user” section and enter the public address of the account; finally, press “Add”.

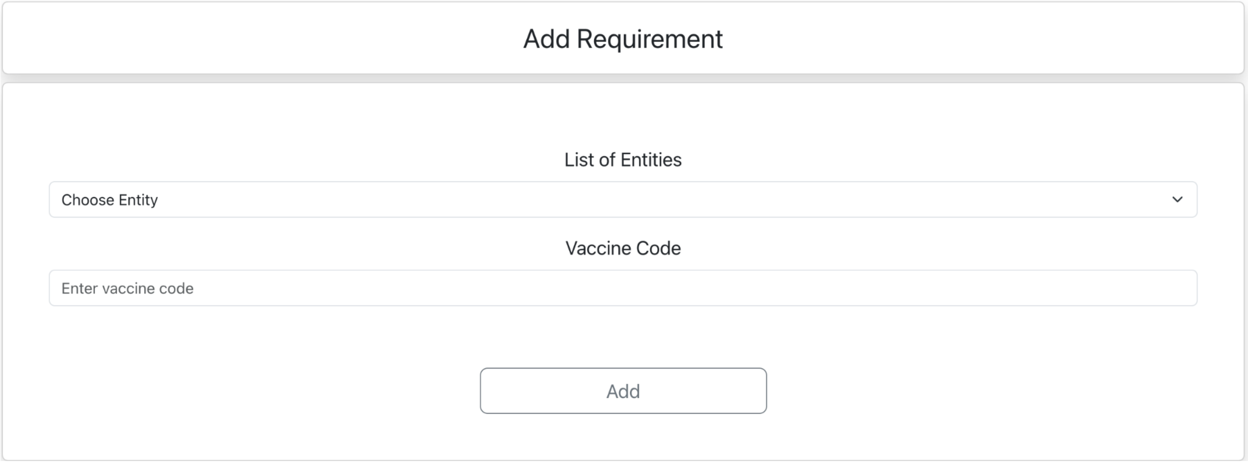


The administrator can also decide to make an account an *entity*. In order to perform this action, scroll to the “Add entity” section; put the public address of the desired account and select a name to best represent that entity. As for vaccine names, the standardization of entities’ names is alike a government concern.



- *Adding a requirement to an entity:*

Scroll down to the “Add requirement” section; then, select the desired entity from the dropdown menu; a list of all existing entities will be shown; finally, add the vaccine code to add to that entity’s requirements and press the “Add” button. Note that if you wrote a non existing vaccine code, the system correctly wouldn’t allow it, notifying the error back to the user.

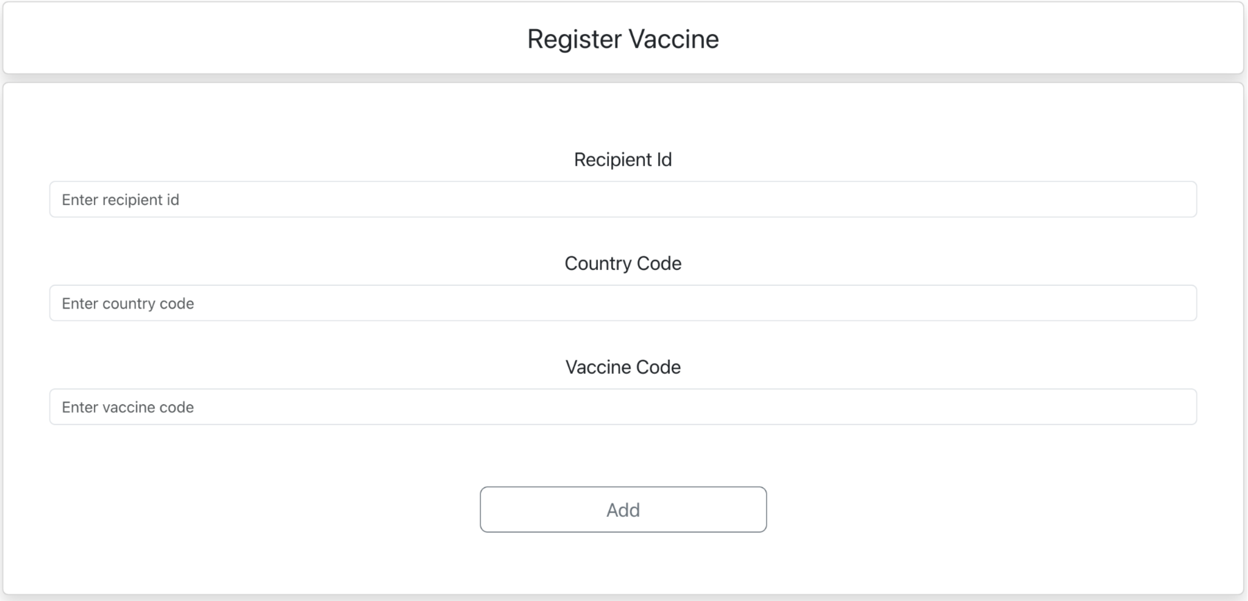


- Actions that can be performed only by authorized users:

In order to perform these kinds of actions, you need to be logged in to your Metamask wallet with authorized user privileges, i.e. with the account that has been granted privileges by the system administrator. In this case, the only possible action is:

  - *Giving a vaccine to a recipient:*

After logging in as an authorized user, locate the "Register Vaccine" card. Then, insert the unique user identifier (following the adopted standard) and subsequently the country code (e.g. "IT", "EN", "DE", etc...) and the vaccine code. The vaccine code must be one approved by the central authority, i.e. it has to be some from the set the admin had created. If the vaccine code does not exist, vaccination registration won't be allowed.



Register Vaccine

Recipient Id

Enter recipient id

Country Code

Enter country code

Vaccine Code

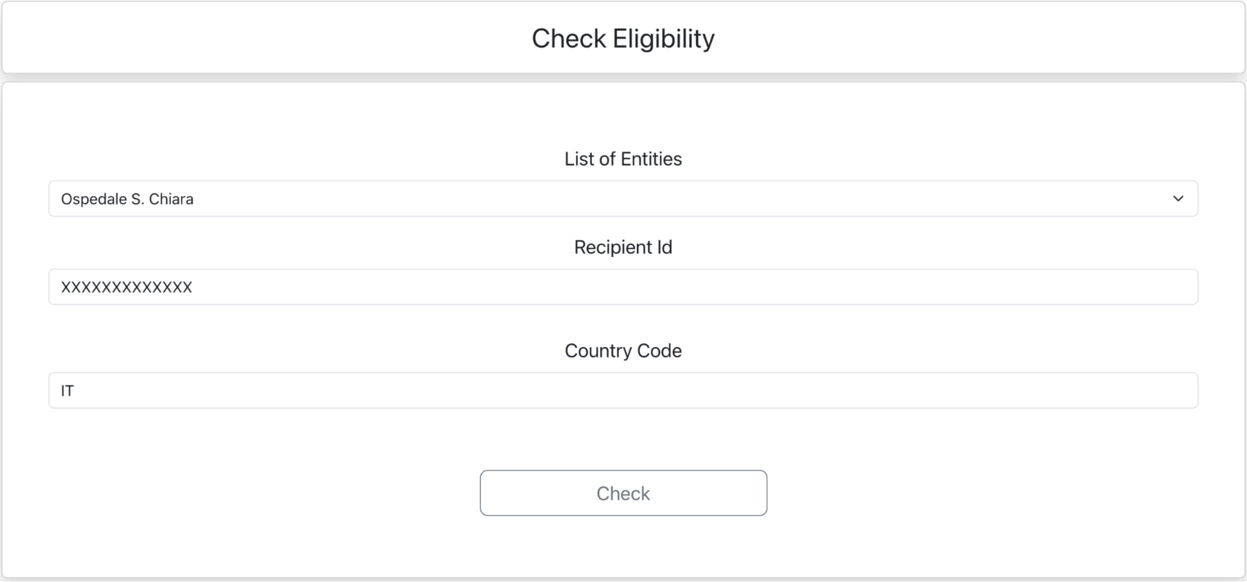
Enter vaccine code

- Actions that can be performed only by both normal users and authorized users:

No authorization is needed to use these functionalities. In this way either an authorized user can check the status of a common user, or a common user can check their own status without the need of extra credentials.

  - *Checking the eligibility of a user:*

To check if the user is eligible to access a certain entity in the list, locate the “Check Eligibility” card. Then, find the entity you wish to see if the user is eligible from the “List of Entities”. Following that, insert the id and the country code of the recipient. Finally, click the “Check” button. The system will return the information.



List of Entities

Ospedale S. Chiara

Recipient Id

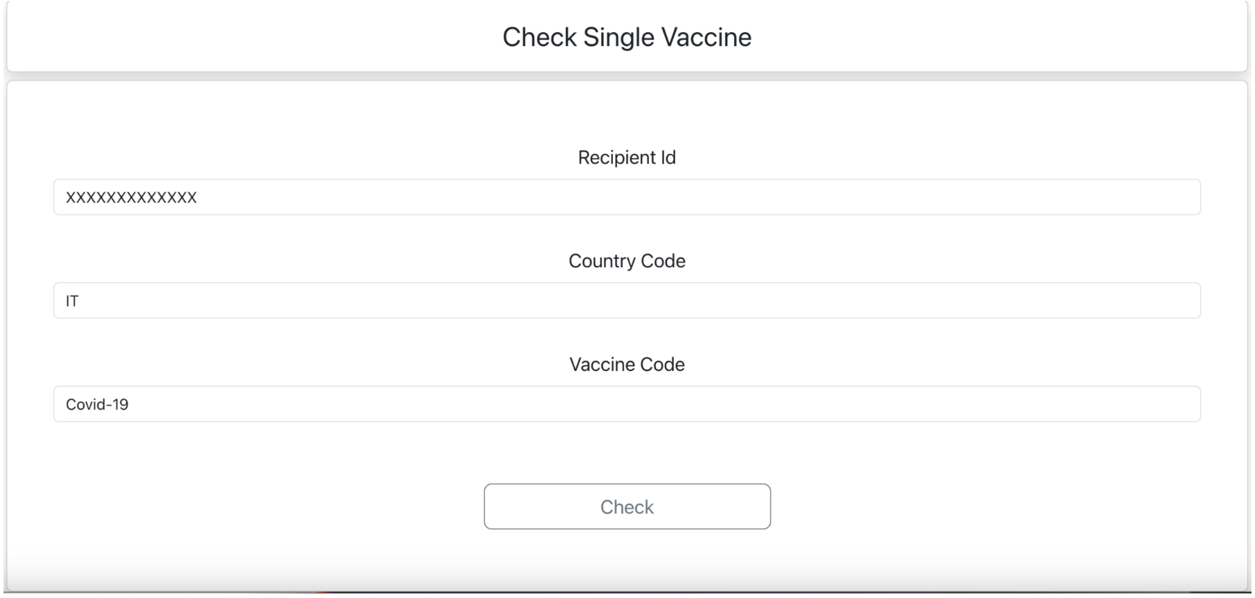
XXXXXXXXXXXXX

Country Code

IT

- *Checking the single vaccine of a user:*

To check if the user has a certain vaccine, locate the “Check Single Vaccine” card. Then, insert the id, the country code of the recipient, and the code of the vaccine that is wanted to be verified. Finally, click the “Check” button. The system will return the information.



Recipient Id

XXXXXXXXXXXXX

Country Code

IT

Vaccine Code

Covid-19

Business Model Canvas

We now outline the various components that constitute the business model of this system. Each component is analyzed in the following list:

Customer Segments

The BBVV System caters to various customer segments, each with distinct needs and requirements:

- Individual Users: These are everyday citizens seeking a secure and convenient method to verify and access their vaccination records. The DApp provides them with a user-friendly interface to check their vaccination status, receive digital certificates, and manage their vaccine history.
- Healthcare Providers: Doctors, nurses, and medical professionals play a critical role in administering vaccines and maintaining accurate vaccination records. The DApp allows healthcare providers to update vaccination data on the blockchain securely, issue digital certificates to patients, and access vaccination histories for medical purposes.
- Government Agencies: Government entities, such as the Ministry of Health, are key stakeholders in the vaccine verification process. The DApp can be utilized by these agencies to ensure the validity and authenticity of vaccination records for citizens, especially in scenarios like international travel or access to public facilities.

Value Propositions

The BBVV System offers several compelling value propositions to its users:

- Enhanced Data Security: By leveraging blockchain technology, the DApp ensures that vaccination records are securely stored and tamper-proof. Users have control over their data, and only authorized entities can access the information, reducing the risk of data breaches.
- Trust and Transparency: The decentralized nature of the blockchain instills trust among users and stakeholders. Each vaccination transaction is recorded transparently on the blockchain, making it verifiable and immutable.
- Convenient Access: The DApp provides a user-friendly interface for individuals and healthcare providers to access vaccination records. Users can easily verify their vaccination status and obtain digital certificates without the need for physical documentation.

- Seamless Integration: The DApp can be integrated with existing healthcare systems and government databases, ensuring interoperability and smooth adoption for healthcare providers and government agencies.

Channels

The distribution and marketing of the BBVV System will utilize various channels:

- Online Platforms: The DApp will be made available on popular online platforms such as app stores and official websites. This ensures easy accessibility for individual users who can download the app on their smartphones or computers.
- Healthcare Institutions: Collaborating with healthcare institutions will allow the DApp to be adopted within medical facilities. Doctors and nurses can directly issue digital certificates to patients using the DApp, creating a seamless user experience.
- Government Partnerships: Establishing partnerships with government agencies responsible for healthcare and vaccination programs will facilitate the adoption of the DApp at a national level. Such partnerships can lead to official endorsements and mandates for vaccine verification using the DApp.

Customer Relationship

Building and maintaining strong customer relationships are crucial for the success of the BBVV System:

- User Support: The DApp will offer comprehensive user support, addressing user queries and concerns promptly. A dedicated customer support team will be available to assist users with technical issues and provide guidance.
- Feedback Mechanism: The DApp will include a feedback mechanism to gather user feedback and suggestions for improvement. This valuable input will be used to enhance the DApp's functionalities and user experience.
- Regular Updates: The development team will ensure regular updates and improvements to the DApp to address emerging needs and challenges.

Revenue Stream

The BBVV System can generate revenue through various streams:

- Government funding: In the hypothetical scenario in which the BBVV is selected as the official country vaccine verification method, generate revenue through public funds and taxes allocated for the development and maintenance of the application.

- Transaction fees: Charge businesses and venues a nominal fee for each verification transaction, covering operational costs.

Key Resources

The successful operation of the BBVV System relies on essential resources:

- Blockchain Infrastructure: Access to the Ethereum blockchain network or other compatible blockchains is essential for storing and managing vaccination records.
- Technology Expertise: Skilled developers and blockchain experts are crucial for the development and maintenance of the DApp.
- Secure Data Storage: Robust and secure data storage solutions are necessary to protect user data and vaccination records.

Key Activities

The key activities involved in operating the BBVV System include:

- Smart Contract Development: Continuously updating and maintaining the smart contracts to ensure accuracy, security, and functionality.
- User Interface Updates: Regularly improving the user interface to enhance user experience and accessibility.
- Integration and Collaboration: Collaborating with healthcare providers, government agencies, and other stakeholders for seamless integration and adoption of the DApp.

Key Partners

Strategic partnerships are vital for the success of the BBVV System:

- Healthcare Providers: Partnering with healthcare institutions and clinics will facilitate the adoption of the DApp within the healthcare ecosystem.
- Government Authorities: Collaboration with government agencies responsible for vaccination programs will lead to official endorsements and widespread adoption.

Cost Structure

The cost structure of the Blockchain-Based Vaccine Verification System includes various components:

- Development and Maintenance: Expenses related to smart contract development, frontend updates, and ongoing maintenance of the DApp.
- Blockchain Transactions: Costs associated with blockchain transactions and gas fees.
- Marketing and Promotion: Investment in marketing and promotional campaigns to increase awareness and adoption of the DApp at national level.

Conclusion

The Blockchain-Based Vaccine Verification System presents a groundbreaking solution to the challenges of vaccine verification. With its secure and transparent approach, the DApp offers immense potential to revolutionize vaccine record-keeping and accessibility while ensuring data privacy and security. By aligning its business model with the needs of various stakeholders, the DApp is poised to make a significant impact on the national healthcare landscape.