Scientific Research Publishing

Journal of Computer and Communications, 2020, 8, 67-80
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Design and Implementation of Highly Secure Residents Management System Using Blockchain

Ragouguelaba Agoda Koussema, Hirohide Haga

Graduate School of Science and Engineering, Doshisha University, Kyoto, Japan
Email: hhaga@mail.doshisha.ac.jp

How to cite this paper: Koussema, R.A. and Haga, H. (2020) Design and Implementation of Highly Secure Residents Management System Using Blockchain. *Journal of Computer and Communications*, **8**, 67-80.
https://doi.org/10.4236/jcc.2020.89006

Received: July 29, 2020
Accepted: September 18, 2020
Published: September 21, 2020

Abstract

This article presents the design and implementation of highly secure and reliable database system for resident records management system using blockchain technology. Blockchain provides highly secure and reliable data access environment. In blockchain, several data fragments are packed into one block and all blocks are connected to form the chain of blocks. In our prototype, each event of resident such as birth, moving, employment and so on, is assigned to data fragment and certain amount of data fragment, says 20 fragments are packed into block. We also developed the web application interface to avoid installing any applications in users' PC or smartphone. Prototype development proved the possibility to use the blockchain technology to large amount of data management system with highly secure and reliable features.

Keywords

Blockchain, Resident Record Management, Database, Data Management, Web Application

1. Introduction

This article presents the design and implementation of highly secure and reliable residents' data management system (RDMS) using blockchain technology. Data management can be defined as the organizing and maintaining data to meet ongoing information lifecycle needs [1]. The importance on data management began with the electronics era of data processing. Data management methods have roots in accounting, statistics, logistical planning and other disciplines before the electronics era. Currently, many countries use electronic data management systems to make people's services more effectively and efficiently.

One of the most important points of RDMS is its reliability and security. Resident data usually include highly confidential personal information such as birth date, postal address, occupation, income and phone number. Therefore, RDMS must be secure, reliable. In order to implement highly secure and reliable RDMS, several equipment and software such as firewall, encryption must be installed. Furthermore, data transmission lines must be protected from illegal access. However, these equipment and software require large cost and operation of these equipment and software also require many highly skilled engineers.

Some countries in the world, especially under-developing countries such as African countries, are still facing the difficulty of data management problem. Such countries do not have enough communication infrastructures such as high-speed data communication lines. Poor communication infrastructure disturbs high-level data exchange system. Furthermore, data management systems must be highly secure, reliable and easy to use. There are already some technologies to implement such highly secure and reliable system. However, many of these technologies require rich IT resources and infrastructure to implement them. Some novel technologies to implement effective data management system under poor communication infrastructure require new idea for realization.

To meet such requirements, the use of blockchain [2] and web application [3] can be one possible solution. Behind blockchain there are several technologies such as cryptography technique and hashing which are used to protect the information. They enable us to implement our final research goal by combining other technology. Blockchain is one of the most promising technologies for our purpose. Web application technology is also essential for under-developing countries. Such countries usually do not have sufficient wired communication networks such as telephone line or high-speed data communication line. This is partly because such infrastructures require huge amount of budget. However recent advancement of wireless communication network dramatically improves the communication infrastructure of such under-developing countries. By using such wireless communication infrastructure such as cell-phone network, people can access websites and get information.

Based on these considerations, we started implementing the highly secure and reliable data management system by combining the blockchain and web application technologies. Many people still misunderstand what kind of technology the blockchain is and how does it work. It enables to implement the secure and reliable control and protection of data on the network systems. If there is intrusions to data, the hacker can access all data and do whatever he wants with it. It is important to have and save secure and reliable data. By coupling these two technologies: blockchain technology and java web framework, we can get a reliable and secure data for a resident data management system. Our system will help to avoid the loss of traceability and falsification of any data. This technology can record the history of all the transaction at any moment by using the chains of data.

One of the most famous applications of blockchain is the application to cryptocurrency. Bitcoin is the most famous example of cryptocurrency which uses

the blockchain technology. And there are already some applications of blockchain technology other than cryptocurrency. For example, applications in the banking ledger [4] [5]. Trading is another promising application field, [6] Identity management is also very impressive application field [7]. Energy and Commodity sector provides new application field of blockchain [8]. Other interesting application of blockchain technology is port logistics or international trading management [9]. These activities include a lot of documents exchanges between several stakeholders. In order to guarantee the correctness and trust, blockchain technology is used. However, there is no work about the management of residents records by blockchain. Therefore, we start designing and implementing the prototype of residents records management system.

We will have the following structure of this paper: In Chapter 2, we will describe the overview of blockchain technology. Chapter 3 explains the system design. Chapter 4 is used to describe the implementation of blockchain framework named Multichain. In Chapter 5, we will mention about the implementation of sample user interface on web browser. Chapter 6 includes the conclusion and future works.

2. Overview of Blockchain Technology

In our system, blockchain technology plays an essential role. Therefore, in this section, we will briefly explain the basic concept of blockchain.

2.1. Definition of Blockchain

The term blockchain is still far from clear to define now. Blockchain itself most likely traces back to Satoshi Nakamoto's original Bitcoin white paper in 2008 [10]. In this paper, Nakamoto described some technology components underlying the cryptocurrency as series of data blocks that are cryptographically chained together. Blockchains are now widely known in cryptocurrency particularly such as Bitcoin. However, there is a big difference between Bitcoin and blockchain. Blockchain is not a Bitcoin. Blockchain is an innovative data management method. Blockchain is the technology used to implement the Bitcoin and other cryptocurrencies. Blockchain was used firstly to design for solving a double spending problem, in other words to establish consensus in a decentralized network. But cryptocurrency is only one of blockchain applications. Blockchain is an authentic distributed database ledger, all of which records the history of the activities along the network among the connected devices or users, which are usually called as nodes. Each node stores the all information on the network with simple secure private keys. The fundamental concept of blockchain is relatively simple; it is a linked list-based database. **Figure 1** illustrates the basic concept of blockchain. There are several blocks in one blockchain. Each block contains some information such as hash value of previous block, data to be stored in the current block and specific value named nonce. Among them, only nonce is controllable, which holds the specific condition of the hash value of each block. Other two are

unchangeable. Hash value of each block is computed using all data in the block. As hash value of each block is computed using all stored data in the block, once some part of data is modified, hash function will generate completely difference hash value and therefore it is very easy to find the falsification of data. In order to generate each block, the user must select appropriate value of nonce in each block and the adjustment of nonce requires huge amount of computing power. Therefore, virtually falsification is impossible. This is the main reason why blockchain provides highly secure and reliable data management system. And blockchain has an attractive function which are very different from the traditional databases. Every node in the network store all blockchain. Therefore, even some accidents such as power failure or cracker's attack destroy the database of one node in the network, all data will be recovered by using the data stored in the other nodes. This is why blockchain has high robust nature. Figure 1 is a conceptual illustration of blockchain database.

2.2. How Blockchain Works

When a user wants to store data in the blockchain, user firstly packs several data fragments into one data package. For example, in case of Bitcoin, each transaction such as transferring money from one user to another is a data fragment. Certain amount of basic data fragments, say 20 fragments, will be packed into one package. After the finish of constructing one package, block construction will start. Each block contains several data such as data package, nonce and previous block hash value. Each block must be connected to existing blockchain. To connect new block to exiting blockchain, hash value of each block must satisfy the specific condition. To satisfy the condition, user must compute specific nonce value. For example, if hash value is represented by 256 bits (32 bytes), hash value of first 64 bits (8 bytes) must be 0. To get such hash value, users will set appropriate nonce value. However, as hash function generates virtually random value, user must repeat computing hash value by assigning specific value to nonce for many times. It requires a huge amount of computing time and power.

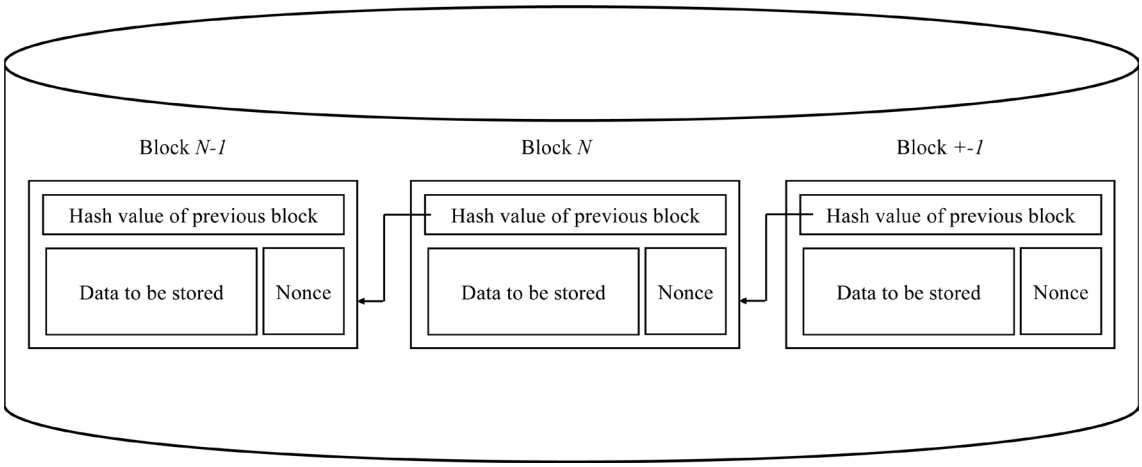

Figure 1. Conceptual illustration of Blockchain.

Once a specific value of nonce which satisfies the condition of hash value is found, newly created block will be connected to existing blockchain.

The reason why blockchain has highly secure and reliability is the amount of computing time. As mentioned above, to connect one block to blockchain, user must compute appropriate nonce value. For example, let n be the length of blockchain (the number of blocks in the blockchain), and the attacker modified the data in *k*-th block. When data in *k*-th block is falsified by someone, hash value of *k*-th block will change drastically. Therefore, the connection of blocks will be broken. If malicious attacker tries to connect modified block to blockchain, he/she has to find new nonce value which satisfies the condition of hash value. The change of hash value of *k*-th block affects following all $(n-k+1)$ blocks; attacker must compute $(n-k+1)$ nonce value again to connect all blocks in the blockchain and this re-computation require quite huge amount of computation. Therefore, falsification of blockchain is virtually impossible. This is why the blockchain system provides highly secure and reliable data.

2.3. Types of Blockchain

There are mainly three types of blockchains: public blockchains, private blockchains and consortium blockchains:

1) **Public Blockchain:** This type of blockchain is opened to public and anyone can participate as a node in the network. The users may or may not be rewarded for their participation. They are not owned by anyone and are publicly open for anyone to participate in. The best example of public blockchain is Bitcoin. Whenever a user transfers a transaction, it is reflected on everyone's copy of the block. Bitcoin uses a public blockchain.

2) **Private Blockchain:** As the name implies, private blockchain is private and is opened only to a consortium or group of individuals or organizations that has decided to share the ledger among themselves. Only the owner makes any changes to it because he has a right. Federated blockchains are faster (high scalability) and provide more transaction privacy.

3) **Consortium Blockchain:** Consortium blockchain is basically a hybrid of public and private blockchains. There are federated blockchains which are operated under the leadership of a group. In a group, there are two or more administrative nodes. Opposite to public blockchains, no one without the approval by administrative nodes is allowed to participate in the process of verifying transactions. For example, in the banking sector, this kind of blockchains are mostly used.

Because of the nature of our target system (resident record management system), public blockchain is not suitable. We will adopt private or consortium blockchain.

3. System Design

3.1. Overall Structure

In this section we will describe the basic structure of our protype system. Our

prototype system uses a blockchain framework named MultiChain [11], Java web application framework PrimeFace [12] and relational database management system MySQL [13]. MySQL is used to store all data of blockchain. **Figure 2** shows the conceptual structure of our prototype.

As shown in **Figure 2**, there are several layers. Bottom-most layer is hard disk or other external storage device. This device stores all data. User will access the stored data by using database management system (DBMS). In our prototype, we use relational DBMS (RDBMS). Blockchain functions are implemented by using blockchain framework. Our prototype system uses MultiChain, which is classified as consortium blockchain. Top-most layer is a framework for web application named PrimeFaces. User interface will be implemented by PrimeFaces framework as a web application. Therefore, no specific application is installed on user's PC/smartphone.

3.2. Mapping Residents Record into Blockchain

Our prototype deals with each resident record as a data fragment of blockchain mentioned in section 2.2. Normally one resident record is generated when a resident was born. Then several update of record will be applied to existing record because of some events such as address change, marriage and so on. And when a resident dead, the dead flag will add to resident record. This is because in blockchain no record can be removed. Instead, dead flag will be added to resident record. As mentioned in section 2.2, it is virtually impossible to modify the data in the blockchain, the correctness of each data in the blockchain will be guaranteed.

3.3. Development of User Interface

We developed our own web application using Java Primefaces framework. It is an open source framework for Java Server Faces (JSF) [14] featuring over 100 components. By using this framework, we will easily develop web application. As it is an open source framework, we can modify to fit other components such as Multichain blockchain framework. As we have implemented our prototype system as a web application, no user needs to implement specific application in their own smartphone or PC.

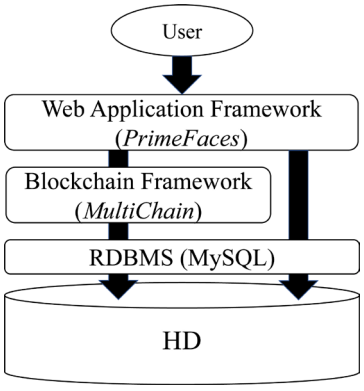

Figure 2. Overall structure of prototype system.

4. Implementation of Blockchain with Multichain Framework

4.1. Introduction to Multichain Framework

Installing Multichain is not difficult. Implementor only needs to follow several instructions for installation. To install Multichain it is required some specific hardware and software requirements. They are the following:

- OS: Linux 64-bit (Ubuntu 12.04+, CentOS 6.2, Debian 7+, Fedora 15+, RHEL 6.2+) or Window 64 bit (Windows 7 or later)
- RAM: Minimum 512 MB
- Disk space: minimum 1 GB

Our prototype system uses on CentOS 7.3 with 2 GB RAM. Firstly, user must download the package of Multichain platform from website (https://www.multichain.com/download-install). At the date and time of this article, user can download version 2.0.7. After downloading the archive, user only need to expand archive file to specific directory in the system. In our prototype system, Multichain is installed on Linux by executing following commands:

```
$ su                  # for changing user into super-user
$ cd /tmp
$ <Downloading multichain archive from website>
$ tar \UTF{2013}xvzf multichain-2.0.7.tar.gz
$ cd multichain-2.0.7
$ mv multichaind multichain-cli multichain-util /usr/local/bin
$ exit               # return to ordinary user
```

4.2. Creating and Connecting Blockchain with Multichain

After installing Multichain platform, user has to create his/her own blockchain into Multichain platform. To create a new blockchain, user must run the following command:

```
$ multichain-util create <name-of-blockchain>
```

If necessary, user can create a clone of existing blockchain:

```
$ multichain-util clone <old-name-of-blockchain>\
          <new-name-of-blockchain>
```

User, of course, can also set any parameter on the command line using the same name, for example:

```
$ multichain-util create <name-of-blockchain>\
            -maximum-block-size=20895656
```

After these settings are finalized, user can start running the blockchain in the background (daemon) with the following command:

```
$ multichaind <name-of-blockchain> -daemon
```

The parameters file params.dat will be locked, initialize the blockchain and create in the first block (genesis). For the first to connect time to an existing blockchain, first user needs to obtain the node address. The node address is shown whenever multichaind starts up or can be get from the getinfo API call. User can connect using the node address as follows:

```
$ multichaind <name-of-blockchain>@<IP_ADDRESS>:<PORT> -daemon
```

In case of private blockchain, user is able to connect immediately because user have not yet been approved permission by an administrator. After permission was approved, user can immediately reconnect to name-of-blockchain using the short form:

```
$ multichaind <name-of-blockchain> -daemon
```

Multichain provides full control over permissions at the network level. There are many global permissions that can granted to addresses connect, send, receive, issue, create, mine, activate, admin. All permissions are assigned on a per-address basis, where addresses can either be public key hashes or script hashes. Permissions can be made temporally by limiting them to a specific range of block numbers. All permissions are granted and revoked using network transactions containing special metadata, which are easy to send using grant and revoke commands for multichain-cli or the JSON-RPC API. The creator of chain's first genesis block has the all basic permissions.

The purpose of blockchain is to create decentralized databases, which are not controlled by any third party. It is necessary to extend the philosophy to the administration of permissions.

5. Implementation Web Service Framework PrimeFaces

We tried to implement our prototype system as a web application. To implement our resident management system as a web application, we combine Multichain blockchain, MySQL database system and Java web framework PrimeFaces.

The data will send and receive from/to user interface through MySQL and blockchain technology. Our system is composed by web application, MySQL database and blockchain technology. The Java web framework is used to insert, update and delete data in the blockchain. In Multichain framework, there is a module which is used to send data to the blockchain technology and MySQL database. To protect data from illegal access from outside of the network, the data will be hashed and encrypted. These data can be sent through internet without any worry. Only the other entity who has a approval in the blockchain network can get the exact information inside the data.

The fundamental module of this system is the web application developed with the java web framework. We test it firstly. A user needs to be authenticated. After the authentication with his role, the user can insert, update or delete an information. The Multichain blockchain module is to be embedded and control the flux of information.

The following test is the communication between the web application and MySQL database on the one hand, in the other hand the Multichain blockchain. At the starting of the web application, we need to authenticate. We must have in mind that; our purpose is to get a registration for births in the national record.

To have a birth certificate at the end, the user needs to have information about Type of hall, Hall name, prefecture, region, information about the child. With the web application, we will have a static of birth by region, by prefecture and by hall.

The test of creating block, creating a consensual governance model and using multiple keys for extra security in Multichain blockchain is an important test work.

5.1. General Description of Java Web Application

Web server is a software that can process the client request and send the response back to the client. Apache is one of the most widely used example web server. Web server is installed on some physical machine and listens to client request on specific port. Web client is a software that helps in communicating with the server. Our browsers can be considered as web client: Firefox, Google Chrome, Safari, etc. when something is requested from server (through URL), web client takes care of creating a request and sending it to server and then parsing the server response and present it to the user. Web server and client uses a common communication protocol, HTTP (Hyper Text Transfer Protocol) which is the communication protocol between server and client. HTTP runs on top of TCP/IP protocol.

Java web application is used to create a dynamic website. Basically, any websites are created with static HTML pages. It means that the user always get the same webpage. However, the information displayed on the screen will be dynamic when web application is used. A web application provides a dynamic extension of web or application server. Web applications are two types: presentation-oriented and service-oriented.

- Presentation-oriented generates an interactive web pages containing various types of markup language (HTML, XML, and ···) and dynamic content in response to requests.
- Service-oriented implements the endpoint of a web service. Presentation-oriented applications are often clients of service-oriented web applications.

Java servlet technology is the foundation of all web application technologies. There are many java web technologies. We can name: Java Server Pages, Java Server Faces, Java Server Pages Standard Tag Library. Java web applications are packaged as web archive (WAR) and it has a defined structure we can export above dynamic web project as WAR file and unzip it to check the hierarchy.

5.2. Sample User Interface

Firstly, when the user accesses the web application, the authentication is required, which is shown in **Figure 3**.

After being authenticated, the user can use the application according to his access rights. In order to access functions, user will use the main page which contains function menu. You can find the selection window in **Figure 4**.

For example, if the user is an administrative user, he/she can create, update or delete data in the database. Figure 5 and Figure 6 are sample interface for updating data in the database.

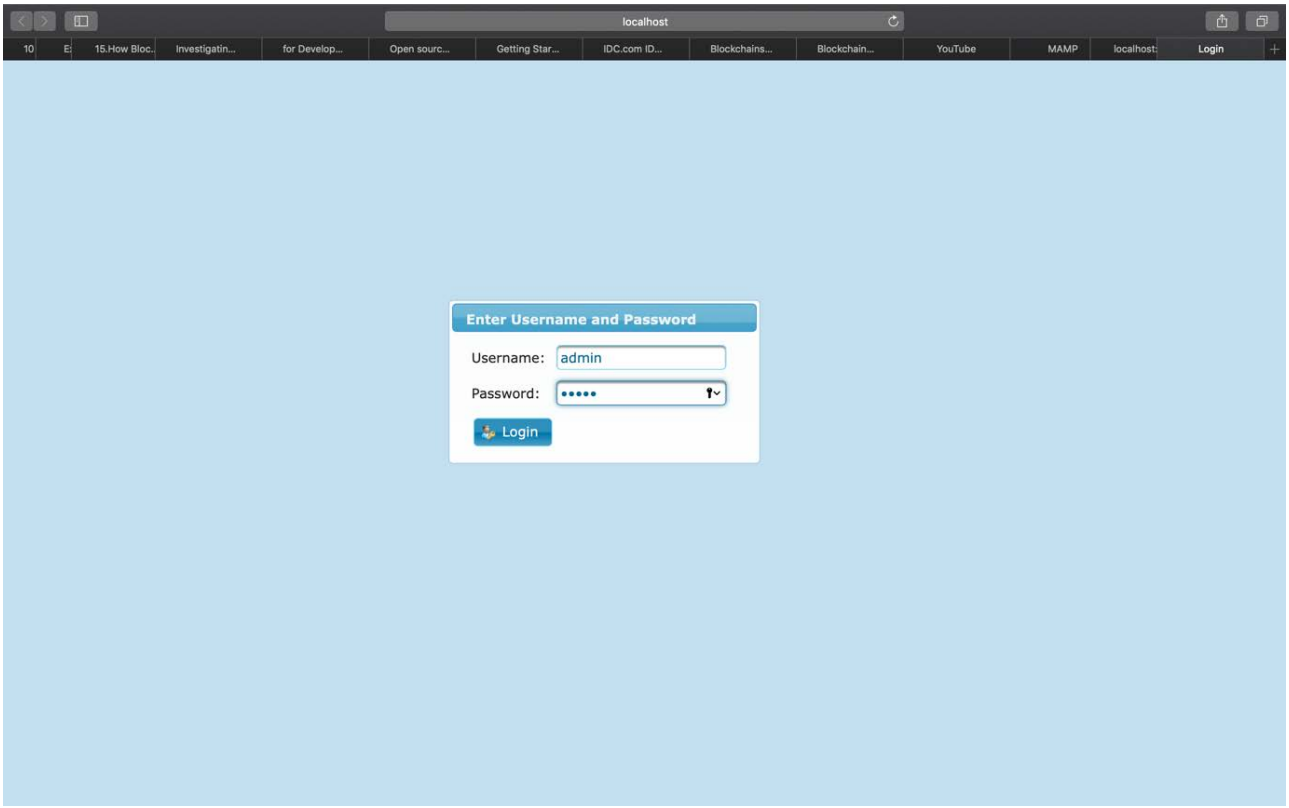

Figure 3. Authentication window.

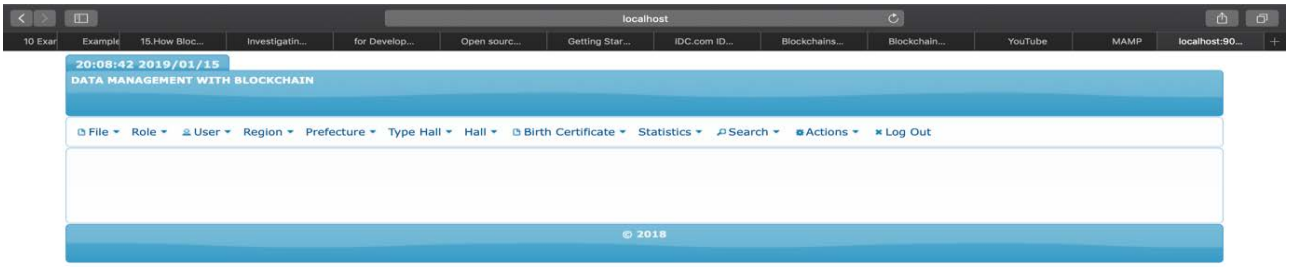

Figure 4. Main menu.

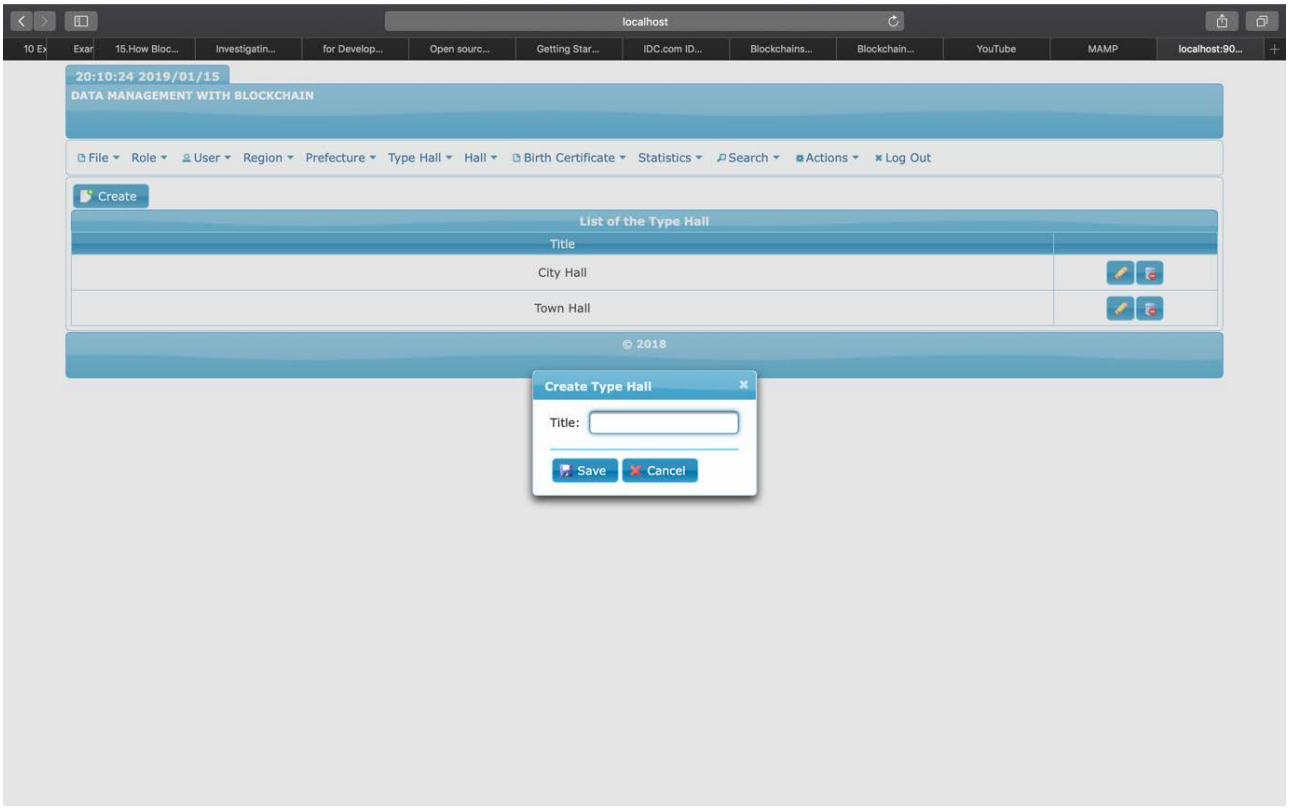

Figure 5. Window for updating data in the database.

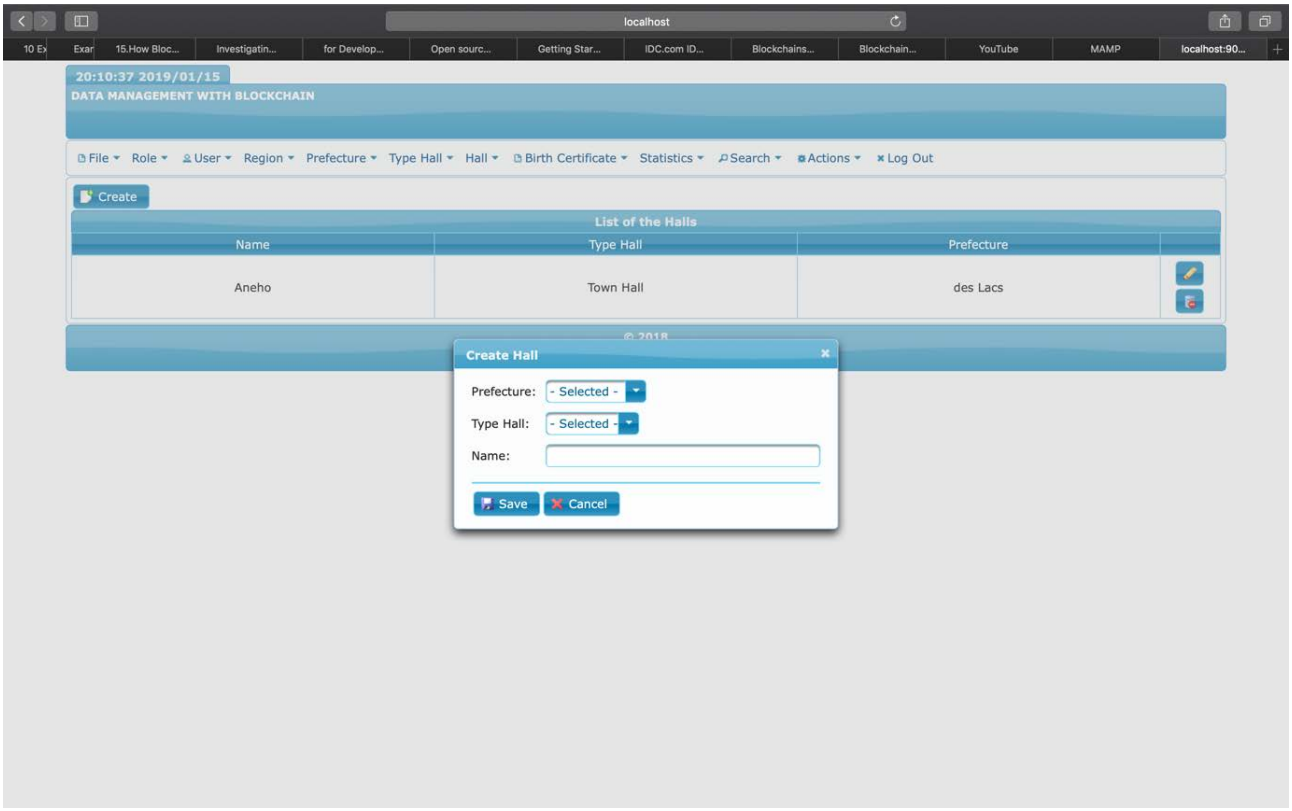

Figure 6. Window for updating data in the database.

After the finish of data registration, end users can get their official record by entering their information through the web. Users enter the information through the interface shown in Figure 7.

User can get, for example, following birth certificate for official use. Figure 8 is a sample official birth certificate document generated by the system.

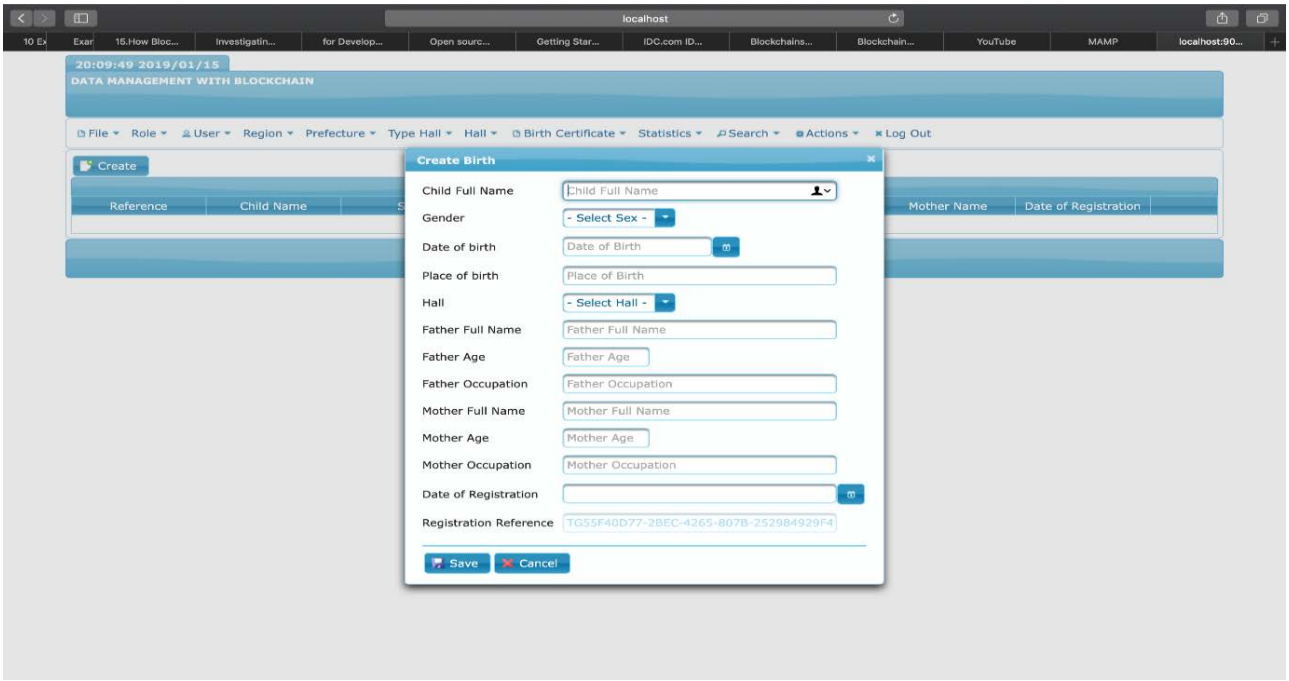

Figure 7. Entering user information to get user's official record.

BIRTH CERTIFICATE

Registration Date: 3/31/18 12:00 AM

Refrence number: TG9995406E-3C35-4FE2-8663-5E4BA7C69C6BTG

Child Name: AGBODJAN Kafui Pelagie

Father Name: AGBODJAN Kodjovi

Gender: Female

Father Age: 36

Date of birth: 3/20/18 12:00 AM

Father Occupation: Farmer

Place of birth: Hilacondji

Mother Name: MESSAN Akpedje

City hall/Town hall: Hilacondji

Mother Age: 24

Prefecture: des Lacs

Mother Occupation: House wife

Figure 8. Generated official birth certificate.

6. Conclusions

In this paper, we described the design and implementation of highly secure and reliable data management system using blockchain. Blockchain technology provides easy implementation of highly secure, reliable data management system. Our prototype probed the efficiency of the blockchain technology to develop highly secure and reliable data management system. Furthermore, we adopted the web application for user interface. Web application approach avoids the installation of any application to users' PCs or smartphones. This approach contributes to the security of the usage of data management.

There are several future works for the deployment of our system. First one is the development of application (APP) for smartphone. Web application is suitable from the security point of view. However, as the size of smartphone display is too small to enter many data. Therefore, to improve the user interface, specific application (APP) is suitable for many users. Another future work is the improvement of processing speed. Currently, to connect one block to exiting blockchain, second order of computation time is needed because of the huge amount of computation to find appropriate nonce value. To process more requests the system, execution time must be faster than now. This will be able to introduce large-scale parallel processing by GPU. GPU provides highly parallel computing.

Acknowledgements

The first author would like to express the financial support from Japanese government named African Business Education Initiative (ABE initiative). The second author thanks to Doshisha University for the support by special research program for COVID-19.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Watson, R. (2020) Data Management—Database and Organizations. eGreen Press, Athens.

[2] Bashir, I. (2018) Mastering Blockchain (2nd Revised). Packt Publishing, Birmingham.

[3] Purewal, S. (2014) Learning Web App Development. O'Reilly Media Inc., North Sebastopol.

[4] Peters, G.W. and Panayi, E. (2015) Understanding Modern Banking Ledgers through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money. https://doi.org/10.2139/ssrn.2692487

[5] Diamond, D.W. and Dybvig, P.H. (1983) Bank Runs, Deposit Insurance, and Liquidity. *The Journal of Political Economy*, **91**, 401-419. https://doi.org/10.1086/261155

[6] Malinova, K. and Park, A. Market Design for Trading with Blockchain Technology.

http://blockchain.cs.ucl.ac.uk/wp-content/uploads/2016/11/Paper_18.pdf
https://doi.org/10.2139/ssrn.2785626

[7] Gail-Joon, A., Moo Nam, K. and Shehab, M. (2008) Portable User-Centric Identity Management. *Proceedings of the IFIP International Information Security Conference*, 573-587.

[8] Dutsch, G. and Steunecke, N. (2017) Use Cases for Blockchain Technology in Energy and Commodity & Trading. http://www.blockchaindailynews.com/attachment/908534/

[9] Mattia, F. (2017) An Explorative Study on Blockchain Technology in Application to Port Logistics. https://www.kennisdclogistiek.nl/publicaties/an-explorative-study-on-blockchain-technology-in-application-to-port-logistics

[10] Nakamoto, S. Bitcoin: A Peer-to-Peer Cash System. https://bitcoin.org/bitcoin.pdf

[11] Coin Sciences Ltd. Multichain for Developers. https://www.multichain.com/developers/

[12] PrimeTek. PrimeFaces Framework. https://www.primefaces.org/gettingstarted

[13] Seyed, M.M. and Tahaghoghi, W.H. (2006) Learning MySql. O'Relilly Media Inc., North Sebastopol.

[14] Geary, C.S. and Horstmann, D. (2010) Core Java Server Faces. Prentice Hall, Boston.