Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Web App Design for Cryptocurrency Transaction System

Semester Project

Xiaochen Zheng
xzheng@student.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zurich

Supervisors:
Ye Wang
Prof. Dr. Roger Wattenhofer

February 26, 2021

Abstract

The project provides a new idea to design a crypto-transaction application. There are two main goals we want to achieve in this project. The first one is that we want provide an easier conversion between physical and digital money. The second goal is to build a pure token-based transaction system. We print digital cryptocurrency into physical notes, which means the physical notes store the value of cryptocurrency directly. In order to achieve these goals, we designed an Android App and a Web App. This report will focus on the design and development of the Web App as well as its function and interaction with database.

1 Background and Motivation

1.1 Background

The few years have seen two new types of digital money emerge. Centralized virtual currencies, usually for the purpose of transacting in social and gaming economies, and cryptocurrencies, which aim to eliminate the need for financial intermediaries by offering direct peer-to-peer online payments[1][2].

Blockchain is the core support technology for the digital cryptocurrency system represented by Bitcoin[3]. The advantage of blockchain technology is decentralization, which enables peer-to-peer transactions, coordination, and collaboration based on decentralized credit in a distributed system. The nodes in the distributed system do not need to trust each other. By applying data encryption, timestamps, distributed consensus, and economic incentives, the blockchain technology can provide a solution to the problems of high cost, low efficiency, and insecure data storage that commonly exist in centralized institutions. With the rapid development and popularity of Bitcoin in recent years, the research and application of blockchain technology have also shown explosive growth. The blockchain technology is considered to be the fifth disruptive innovation in computing paradigm after mainframe, personal computer, Internet, and mobile/social network, and the fourth milestone in the history of human credit evolution after blood relatives' credit, precious metal credit, and central bank bills credit[4]. Blockchain technology is the prototype of the next generation of cloud computing, which is expected to completely reshape human social activities like the Internet and transform from the current Internet of information to the Internet of value.

The cryptocurrency transaction system (also known as the cryptocurrency/digital wallet) is a tool to interact with Blockchain. It is an application, program, or service to store the public and/or private keys for cryptocurrency transactions[5]. Digitalization has become widely popular in society, especially among low-income communities[6]. Digital wallet has the advantages of keeping records of transactions, delayed payments, and security of money. It also allows the creation of payment histories for individuals that can be used to assess their financial credibility.

However, concerning the issue of managing easy daily payments, the debate arises about which payment mechanism - cash or digital - is better[7]? Although one of the advantages of digital cryptocurrencies is that they guarantee the removal of the physical constraint of 'attendance' that comes with cash or physical notes, sometimes this point can also become the opposite side. Since the network does not cover everywhere, how do people make the payment when they don't carry any cash and don't have an Internet connection? Or what to do if the mobile device runs out of battery. Meanwhile, research on how to design digital transaction systems tends to treat

payments as a transactional matter, and usability concerns tend to focus on balancing transaction speed with security[8][9]. Ferreira et al.[10][11] showed that the speed of the transaction was not the only concern of users, because of the social context and social relationships in which the transactions take place to have consequences for the user experience. The trustworthiness of payment systems, whether cash or digital, are socially constructed[12]. Furthermore, some features of the digital transaction system make it less trustworthy than cash to people[7] like the non-tangibility of exchange.

Kaye et al.[13] proposed suggestions on how to make digital payments more trustworthy based on the user's experience. It is shown that to some extent, the emotional components of the decisions that people make around money dominate the user's preference. While some of the decisions people make may be irrational from a purely financial perspective, they are a combination of personal feelings and values. Personal aspects should be incorporated into the financial domain when designing transaction systems. Meanwhile, users should feel secure when the system is used to track their transactions and manage their assets. That inspire us to design systems based on not only the financial rules but also the real world daily practices, rather than idealized concepts for optimizing financial transactions. Considering these aspects, making digital cryptocurrencies physical would be feasible. There are two related work about physical cryptocurrencies.

The Bristol Pound is a 'mixed-media' currency: payments can be performed using physical printed notes, online via a browser, and using a platform independent text messaging system on mobile devices[10]. But users can print notes only from a limited number of facilities in the city.

The other instance is Kong - an instance of physical cryptocurrencies[1].

> *'Kong is a hybrid cryptocurrency that consists of both physical notes and digital tokens. Transactions in physical Kong notes are offline. They do not require any accompanying digital transaction to be valid and they are immune to factors that adversely effect on-chain transactions like spikes in transaction costs or network congestion.'*

1.2 Motivation

With the rise of digital payments in recent decades, people are becoming more and more accustomed to electronic payments. But cash still accounts for a large portion of global transactions. Currently, to withdraw or deposit money, we must go to a bank counter or an ATM. But this is inconvenient and even unsafe due to the current pandemic. Therefore, we hope that our transaction model can avoid this inconvenience.

[1] https://kong.cash/

Considering the means of payment people are used to, this inspires us that the combination of physical and digital cash is more effective and efficient in real life. There are two main goals the project wants to achieve in this project. The first one is that we want provide an easier conversion between physical and digital money. The second goal is to build a pure token-based transaction system. We want to make an analogy of the cash we used for thousands of years. We aim to valid cryptocurrency only based on the token itself, not the payer. Meanwhile we plan to design the cryptocurrency which can be verified offline.

In this report, the design and development of Web App are mainly introduced. We developed a Web App which achieved most of the function of our cryptocurrency transaction model via web pages: interacting with the central server and database, checking account information, making the account-to-account transaction, generating tokens and storing the tokens in the form of QR code, storing the QR code in a PDF file, printing digital cash. As a complement of the Android App, the Web App can be accessed in all platforms such as PC, laptop, iOS mobile phone, and Android mobile phone.

2 Project Overview

2.1 Structure

As shown in Fig.1, the model achieved two goals mentioned in section 1.2 by building Android App and Web App which can be connected to a centralized database and a server. The database stored all users' information and the tokens and addresses of transactions. Transactions are conducted by interacting between App and database/server. Besides, a Bluetooth printer and NFC chips are needed for printing money.

2.2 Token-based Cryptographic Transaction System

The main function of token-based cryptographic transaction system is to generate the physical token for transaction. A one-time used token address and an asymmetric key pair are needed. The one-time used token address is generated randomly by the server. The key pair is generated by applying RAS encryption algorithm. The algorithm is shown in Algorithm 1: first choose two different large prime numbers p and q at random and calculate $N = p \times q$; then calculate $\varphi(N)$ where φ is the Euler's totient function; choose an integer e which is smaller than $\varphi(N)$ and mutually prime with $\varphi(N)$; calculate the modular multiplicative inverse of e with respect to the modulus $\varphi(N)$; obtain the public key and secret key by (e, N) and (d, N) respectively.

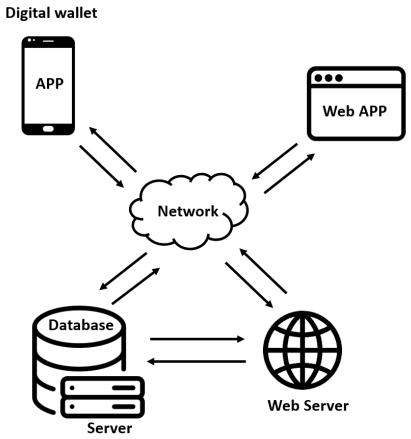

Figure 1: Project structure

Algorithm 1 RSA Encryption

procedure KEY GENERATION
    Choose two different primes p, q.
    $N = pq$.
    $\varphi(N) = \varphi(p)\varphi(q) = (p-1)(q-1)$, if N is prime, $\varphi(N) = N - 1$.
    Choose encryption exponent e *s.t.* $gcd(e,\ \varphi(N)) = 1$.
    decryption exponent d *s.t.* $ed = 1(mod\varphi(N))$.
    Output $pk = (e, N)$, $sk = (d, N)$.
procedure ENCRYPTION
    $c = m^e\ mod\ N$.
    $m = c^d\ mod\ N$.

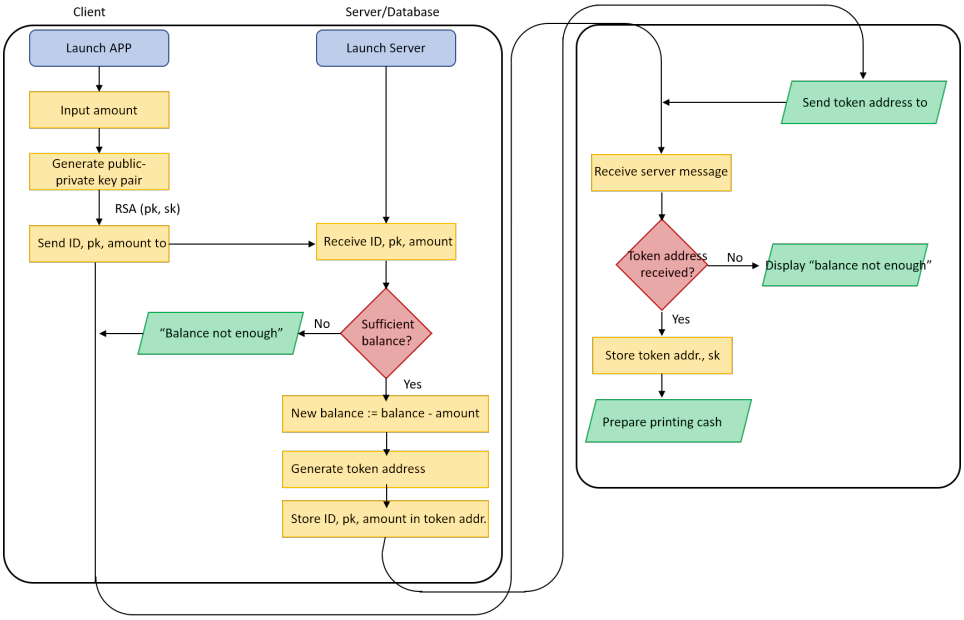

Figure 2: the process of generating token

The process for generating physical token is shown in Fig.2[2]. When a transaction is requested by user, the App will generate the RSA key pair locally and send the public key together with the required amount to the server. The server will check the sufficiency of balance. With the sufficient balance, the server will generates a random token address and store the amount and the public key in this address. After all operations, the server will send the token address back to the user. The physical token in the form of QR code is illustrated in Fig.3

3 System Design

This section will show the design and development of the entire system of the Web App and Web Server which are served as complement of the Android App.

3.1 High-Level Architecture

In order to develop a Web App and Web Server for the cryptocurrency transaction model, we applied a three-tier architecture, which is a kind of client-server architecture for the usage of web development. It consists of

[2]Inspired by Yu Chen, yuchen14@student.ethz.ch

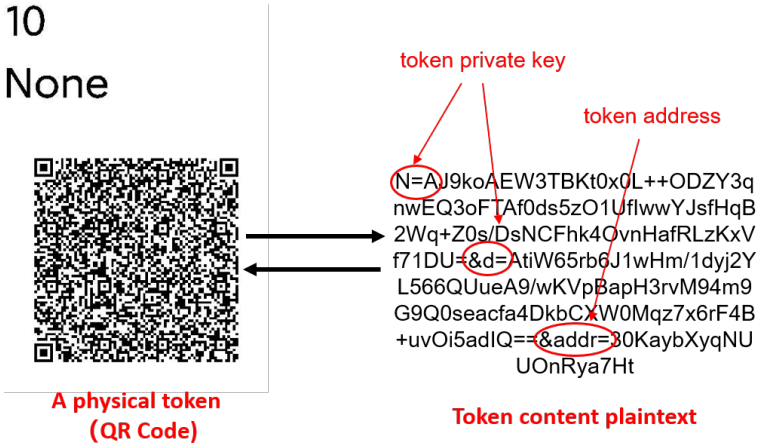

Figure 3: the physical token in the form of QR code

a presentation layer, a business layer, and a persistence (data) layer[3]. The communication flows between the layers are represented by[4]:

```
Presentation <-> Business <-> Persistence
```

Presentation Layer is the topmost level of the application which contains the user interface(UI) and presents information to the user by building HTML pages and receiving user input through web forms requests. The Servlet's and JSP's combination is applied.

Business Layer contains all logic tied to the web application functionality which controls an application's functionality by performing detailed processing. The functionality of the business layer in this project is (1) initiating the transaction; (2) creating/deleting a new user account.

Persistence Layer includes the data persistence mechanisms (database servers, file shares, etc.) and the data access layer that encapsulates the persistence mechanisms and exposes the data which performs create, read, update and delete persistence operations, etc.

3.2 Architecture Details

The detailed architecture is shown in Fig.4. The functions of each component are:

[3]https://en.wikipedia.org/wiki/Multitier_architecture#Three-tier_architecture

[4]https://stackoverflow.com/questions/286846/describe-the-architecture-you-use-for-java-web-applications

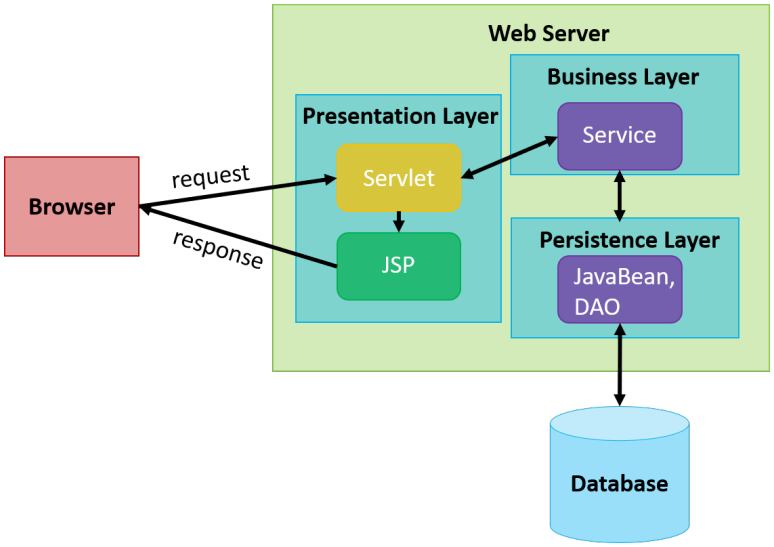

Figure 4: the architecture of Web App

JSP for presenting and collecting input data.

Servlet for validating data, instantiating JavaBean, calling DAO to connect to the database, controling pages jumping.

Servlet filter for processing the request before it reaches the servlet, and the response when it leaves the servlet.

Servlet listener for listening to various operations of the Web App, and handling the events generated when the relevant operation is triggered.

DAO for connecting to the database and database operations such as: query, delete, modify, etc.

JavaBean for data encapsulation and transferring the query results between servlet and JSP pages.

4 Implementation Details

This section will introduce the detailed implementation methods which are used to build the Web App and Web Server. The presentation layer details contain the design of JSP code and web content. The details of business layer as well as the persistence layer will also be shown as the implementation of several user-defined Java packages. The procedure of the system will also

be described.

4.1 Java Server Pages (JSP) and Web Content

The Java Server Pages (JSP) is applied to develop the Web App. There are some feasure about JSP. (1) JSP is a dynamic web development technology. It uses JSP tags to insert Java code into HTML pages. The tags wrap the Java code with `<% ... %>`. (2) JSP is a Java servlet that is mainly used to implement the user interface part of a Java web application. JSPs are written by combining HTML code, XHTML code, XML elements, and embedded JSP operations and commands. (3) JSP fetches user input data and accesses databases through web forms, and then dynamically creates web pages. (4) JSP tags have a variety of functions, such as accessing databases, recording user selection information, accessing JavaBeans components, etc. They can also send and share information among different web pages.

/Server/WebContent/login.jsp: It allows users to input the username and password. It will pass these two objects to `check.jsp` for further verification. It also contain the url to download the full user guide. After clicking the "Login" button, it will sent the username and password to `check.jsp`.

/Server/WebContent/check.jsp: It inherits/gets the parameters from the input of `login.jsp` and execute querying the validation of those combination in the database. If there is no such combination in every entry of the database, it will return "wrong password" or "username does not exist" depending on the circumstances. This is achieved by calling the class from `Bean.DBBean` package (which will be introduced in section 5.3). The two objects will compose the SQL querying sentences which will be executed via calling function in `Bean.DBBean` package.

/Server/WebContent/transaction.jsp: This web page will achieve the account-to-account transactions. Users are asked to input the payment account, payee account, and amount. It will first import two user-defined packages: `service.UserService` and `util.DBUtil`. After user clicks the "Transfer" button and the function successfully makes the connection between central server and database, the function `userService.transfer()` will be called and the required parameters will be passed into it.

/Server/WebContent/QRcodeInput.jsp: It generates the token and stores the token into a new text by importing the `Bean.Crypto`. It first parse all required parameters inheriting from the former input and store them in object `value`. With the function `crypto.generateQRcontent()`, the private key, address, and modulus will be obtained and be stored as a `ciphertext`. Then the `ciphertext` will be sent to `QRcode.jsp`.

/Server/WebContent/QRcode.jsp: It generates the QR code containing the private key, modulus, and address. It creates a new empty bit matrix with width and height of 300 pixels. Then the tokens will be encoded and be written into the bit matrix which will be stored as a gif file.

4.2 Data Access Object (DAO)

The data access object (DAO) is a pattern that provides an abstract interface to some type of database or other persistence mechanism. By mapping application calls to the persistence layer, the DAO provides some specific data operations without exposing details of the database[5]. DAO design pattern provides a general pattern to simplify code and enhance the portability of the program.

In our project, JavaBean plays the same role as DAO. We also implement some component in the `Bean` package. DAO consists of 5 important parts: Database Connection Class, Value Object(VO), DAO interface, DAO Implementation Class and DAO Factory Class.

Database Connection Class A Java class for connecting to the backend database. The Java Database Connector (JDBC) driver was applied. There are two steps to make the connection: (1) register the driver class, (2) create the connection object and statement object. With the connection, queries with the statement object can be executed. Four parameters are required to make a database connection: database driver class `DBDriver`, database connection `URL`, username, and password.

VO is a Java class corresponding one-to-one with the database table. Contains properties that correspond one-to-one with the database table fields. VO provides an object-oriented method to manipulate the database. The DAO interface performs database operations by calling VO. The VO is store in `src/entity/UserInfo.java`.

DAO interface defines all user operations, such as writing, deleting and querying records.

DAO Implementation Class implementation of DAO interface and corresponding methods.

DAO Factory Class to get the specific DAO implementation class.

[5]https://en.wikipedia.org/wiki/Data_access_object

4.3 Java Packages and Procedures

We write several Java packages to build the Web App and Web Server. Those Java packages are main content of this part. The detailed code map consisting of Java packages and Java file is shown in Fig.5.

Server/src/Bean/Crypto.java: This generates the public-private key pair to print the physical cash based on the RSA algorithm as mentioned above. The key is stored in the token and printed as QR code. The Java code generateing public-private key pair is as follows:

```
keypair = RSAUtils.generateRSAKeyPair(512);
RSAPublicKey publicKey = (RSAPublicKey) keypair.getPublic();
RSAPrivateKey privateKey = (RSAPrivateKey) keypair.getPrivate();

/* parameter e */
pk_exp = Base64Utils.encode(publicKey.getPublicExponent().toByteArray());
/* parameter d */
sk_exp = Base64Utils.encode(privateKey.getPrivateExponent().toByteArray());
/* parameter N */
modulus = Base64Utils.encode(publicKey.getModulus().toByteArray());
```

Server/src/Bean/DBBean.java, Server/src/util/DBUtils.java: These are served as part of the DAO function. Use the Java Database Connector (JDBC) driver, an API defining how a client can access a database, to establish the connection between the central server and the MySQL database.

Server/src/dao/UserDAO.java: It contains the basic functions of the model.

Logincheck: It takes username and password as parameters. First this procedure start the connection between the central server and database. Then certain entry from the database is selected and is used to verify the right matching between username and password. If those combination exists, the detailed information of the account will be returned. Otherwise 0 will be returned and it shows that the username or password is wrong or the account does not exist.

Displayinfo: After checking the login information, the detailed information consisting of balance, email, telephone will be shown on the web page. This procedure is achieved by fetching the whole entry (certain account) of the database.

Exetransaction: The procedure achieves the accout-to-account transaction function. Users need to input the payee account, the payment account, and the amount. It will first verify the payee and payment accounts. If the

payee account is invalid, it will return `TYPE_ID_FAILED`. If the payee account and the payment account are the same, it will return `TYPE_SELF_FAILED`. After verification, this function will check the account balance. If the account balance is insufficient, the procedure will be interrupted and the function will return `TYPE_AMOUNT_FAILED`. Only when all rules are followed, the money will be transferred successfully and the function will return `TYPE_TR_SUCCESS`.

Cryptotransfer: This procedure is used to generate the tokens to be stored as a PDF file via web page. It takes address, value, account id, N (modulo) and public key as parameters. The PDF saving process is achieved by `window.print()` in `QRcode.jsp`. If the tokens are successfully created, the balance of the payment account will be updated and a new entry containing address, value, N, public key and crpto_time will be inserted into cryptotransferdb[6].

In addition to what we have mentioned above, it is important to emphasize the **`service`** package. It contains **`/Server/src/service/UserService.java`** which defines several operation functions such as (1) showing account info, (2) transferring money account-to-account, (3) generating tokens to cryptotransfer by calling function from **`UserDAO.java`** (DAO layer). The function of showing the account information is obtained by implementing:

```
1  package service;
2
3  import java.sql.Connection;
4  import java.sql.SQLException;
5  import java.util.List;
6
7  import dao.UserDAO;
8  import util.DBUtil;
9  import entity.UserInfo;
10
11 public class UserService {
12    UserDAO userDAO = new UserDAO();
13    public String login(Connection conn,String username, String password) {
14       UserInfo userInfo = new UserInfo();
15       userInfo.setUsername(username);
16       userInfo.setPassword(password);
17       return userDAO.logincheck(conn,userInfo);
18    }
19
20
21    public Object[] accountinfo(Connection conn,int account_id) {
22       return userDAO.displayinfo(conn,account_id);
23    }
```

[6]This is developed by Shenyi Wang, shewang@student.ethz.ch

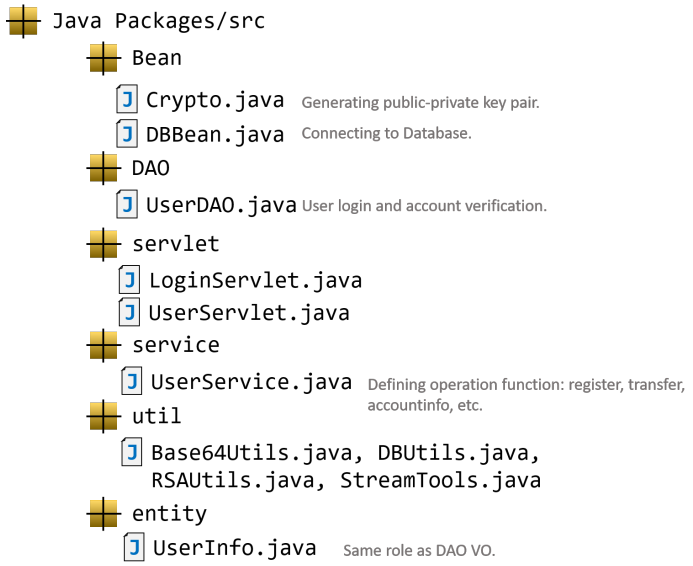

Figure 5: Java package and Java file

In the **`UserService.java`**, it is achieved by calling `userDAO.displayinfo()`.

4.4 Function and User Interface (UI)

Fig.6 shows the function and UI of the Web App. After successfulf login, the detailed account information is shown and there are two functions users can choose: generating QR code for printing digital money and directly transferring money to other users. For generating QR code, a sufficient amount of money needs inputting into the input field. Then the QR code can be downloaded as PDF format. For the online transaction, users need to enter the sufficient amount and payee's ID. How the Web App interact with server and database is similar with the method introduced in Sec.2.

5 Result and Discussion

The Web App can be accessed via

http://82.130.102.183:8080/WebServer/login.jsp

The Web App is a simplification and complement of the Android App, focusing only on the user's functions. For the merchant who wants to have the functions the same as Android App such as returning change to customer, canceling the payment at any time, changing the price at any time, more work is needed to develop a Web App supporting offline payment. However this is a good try to explore applying our idea to different platform.

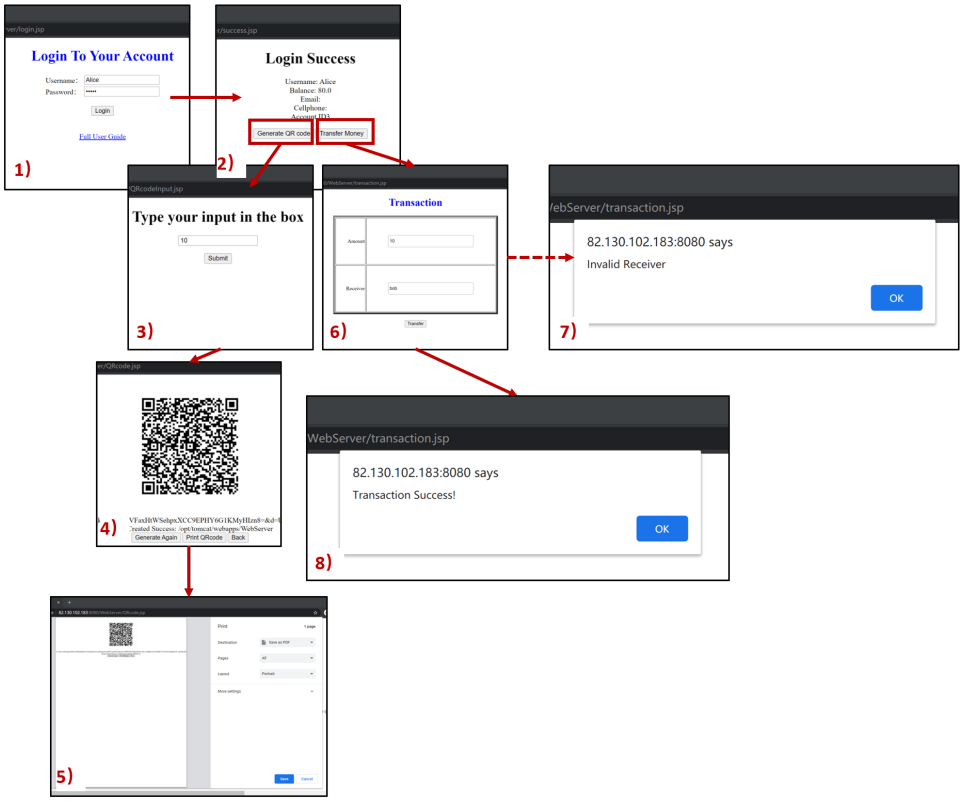

Figure 6: the operation function and UI of the Web App

References

[1] Peters, G., Panayi, E., Chapelle, A.: Trends in cryptocurrencies and blockchain technologies: A monetary theory and regulation perspective. In: Journal of Financial Perspectives. Volume 3. (November 2015)

[2] Chaum, D.: Blind signatures for untraceable payments. In: Advances in Cryptology, Springer, Boston, MA (1983)

[3] Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot (2019)

[4] Swan, M.: Blockchain: Blueprint for a new economy. O'Reilly Media, Inc. (2015)

[5] Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. Ieee Access **4** (2016) 2292–2303

[6] Mehmood, H., Ahmad, T., Razaq, L., Mare, S., Usmani, M.Z., Anderson, R., Raza, A.A.: Towards digitization of collaborative savings among low-income groups. Proc. ACM Hum.-Comput. Interact. **3**(CSCW) (November 2019)

[7] O'Neill, J., Dhareshwar, A., Muralidhar, S.: Working digital money into a cash economy: The collaborative work of loan payment. Computer Supported Coop Work '26 (2017) 733–768

[8] Balan, R.K., Ramasubbu, N., Prakobphol, K., Christin, N., Hong, J.: mferio: the design and evaluation of a peer-to-peer mobile payment system. In: Proceedings of the 7th international conference on Mobile systems, applications, and services. (2009) 291–304

[9] Lehdonvirta, V., Soma, H., Ito, H., Yamabe, T., Kimura, H., Nakajima, T.: Ubipay: minimizing transaction costs with smart mobile payments. In: Proceedings of the 6th international conference on mobile technology, application & systems. (2009) 1–7

[10] Ferreira, J., Perry, M.: Building an alternative social currency: Dematerialising and rematerialising digital money across media. In: Proceedings of HCI Korea. HCIK '15, Seoul, KOR, Hanbit Media, Inc. (2014) 122–131

[11] Ferreira, J., Perry, M., Subramanian, S.: Spending time with money: From shared values to social connectivity. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work Social Computing. CSCW '15, New York, NY, USA, Association for Computing Machinery (2015) 1222–1234

[12] Wolman, D.: The end of money: Counterfeiters, preachers, techies, dreamers–and the coming cashless society. Hachette UK (2013)

[13] Kaye, J.J., McCuistion, M., Gulotta, R., Shamma, D.A.: Money talks: Tracking personal finances. CHI '14, New York, NY, USA, Association for Computing Machinery (2014)