IEEE Conference on E-Commerce Technology, July 6-9, 2004, San Diego, California, USA

On Designing a Flexible E-Payment System with Fraud Detection Capability

Antoinette Leung
Department of Electronic Commerce
Companhia de Telecomunicacoes de Macau
Macao
ant@qaisp.net

Zhuang Yan, Simon Fong
Faculty of Science and Technology
Universidade de Macau
Macao
{syz, ccfong}@umac.mo

Abstract

Nowadays security measures especially fraud detection has become an imperative issue in the design of online payment systems. In addition to standard network security schemes such as data encryption and firewalls, payment systems with internal logics that are capable of detecting fraud transaction, and even potential fraud transactions are preferred by both users and solution providers. Currently on the market one can easily purchase a software package for implementing an e-payment system with most standard payment functions. Fraud detection capability, usually implemented as an extra add-on internal logic in the system, however is not included in most packages. It would have to be built in-house. This paper sheds some light on the designing issues on this add-on fraud detection module, namely Fraud Detection Manager. The design is based on the concept of atomic transactions called Coupons that we implemented in e-wallet accounts. This is a relatively simple approach to deter fraud and resolve dispute without the use of digital signing [1]. Some issues in handling fraud transactions are discussed from our experiences in operating a commercial e-wallet service.

Keyword: e-Wallet architecture, Fraud detection

1. Introduction

In current market place, there are quite a lot of different payment solution providers. Most of them come with banking background as they are in comparative advantageous position to effect payment transfers amongst payers and payees given the support of banking network. Yet due to the fact that banking network is complicated, and there are many commercial reasons for banks around the world to integrate and connect, third party payment solution providers still have a role to play.

Given the popularity of the credit card market, payment by credit card is the most inclined payment method. Payment by account transfer is restrictive as payer and payee may have different bankers, which do not have direct relationship. Merchants do not favor delay of payment as a result of inter-bank transfer. Payment by other means like authenticated e-Cash require tremendous investment on both technology and operating cost. User-friendliness and user-acceptance are key obstacles for e-Cash to gain popularity in a near future. Therefore for commercial feasibility, at least as for today, it is desirable to build an e-payment system similar to PayPal but with emphasis on enhancing the credit card payment method. That is, an e-Wallet system that runs on an intermediate party between the users and merchants. Users can create an e-Wallet account that accepts fund-in from users via direct bank transfer, debit cards as well as credit cards.

Users enjoy a peace of mind in using the e-Wallet service during on-line shopping without the need to surrender their credit card details on merchant websites. They can choose to top up their e-Wallets using direct transfer or credit card payment whereas notational money from their e-Wallets is debited at each purchase.

The advantage is leveraging the well acceptance of credit card as an e-Wallet fund-in method for the end-users. This however implies a problem to the e-Wallet operator in case of any credit card related dispute raised by the user that has to be resolved by the e-Wallet operator instead of by the merchant alone. In other words, the responsibility of handling fraud transactions and subsequent dispute is shifted from the merchants to the e-Wallet operator.

From our working experience in running an e-Wallet service provider 1-Pay.com, building an e-payment system is not technically difficult since the technology is widely available off the shelf. (Computer hardware and e-Payment software toolkits can be purchased, programmers can be employed.) The challenge is taking preventive measures to tackle frauds.

In the latter sections of this paper, we discuss about possible frauds pertaining to e-Wallet transactions and how do we deal with them both logically and technologically. We designed a generic fraud-detection module that can be configured with rules in governing the meticulous level of checking. The internal design of the e-Wallet account would have to be modified with the concept of Coupon-based transaction for enabling traceability. The design of the fraud-detection module and the modification of the e-Wallet account are made flexible enough to integrate and inter-operate with most existing e-Payment systems as an add-on.

2. Credit Card Frauds in E-Wallet Services

In our e-Wallet model where token money is prepaid by the users via debit cards or credit cards, and thereafter the money can be spent at the merchant websites for purchases, it requires a fraud-free environment so to attract users and merchants using the e-Wallet service. User acceptance is a fundamental requirement for third-party payment provider to survive.

Fraud in online business is inevitable. Frauds in e-Wallet services that are mainly on credit cards, come usually in three different forms and have two degrees of damage such as financial loss and reputation blemish.

One type of fraud is that a stolen credit card number is used for making a purchase; and the other one is that the user used a legitimate credit card but deliberately denies the transaction upon receiving the goods or service from the merchant; the last type is from the dishonest merchant who charges the user's e-Wallet without consent.

In any case, a user would raise a transaction dispute through the credit card issuer bank. Some common transaction disputes are as fellow [2]:

- "I agree to pay $X, but was charged $Y instead"
- "I've only bought $X worth of goods, but my e-Wallet balance has gone down by $Y"
- "I never bought this, but it appears on my account statement"
- "I told the merchant no, but he put it through anyway"

Upon any transaction dispute brought up, the e-Wallet service provider would have to answer, by at least showing records of the transactions.

If the dispute transaction was made by an e-Wallet that is funded in purely by debit card or cash, the resolution would be handled (or argued) between the user and the merchant. Sometimes, the e-Wallet operator would have to reverse-debit the wallet account provided that the merchant refunds the user money.

However, if the e-Wallet account is funded by credit card payment by the user that is a very usual case, things become trickier. Because the e-Wallet account serves as a common pool of notational money merging from different forms and sources of payment, it would be almost impossible to separate out which amount of money that come from which credit card account was involved in the transaction dispute. For example, a credit card owner files a transaction dispute via his credit card issuer that $1000 was mistakenly debited by our e-Wallet provider. Checking on the e-Wallet account under dispute, it was found that the initial balance was $200 funded in by cash. Now plus the $1000 fund-in from credit card, a total of $1200 from the e-Wallet was spent over three different merchants A, B and C each with a transaction of $400. In this scenario, how would the e-Wallet provider approach the three merchants about the fund-in dispute of $1000? There is no way to know in which merchants the $1000 was actually spent let alone the proportion of it. (It could be $400 in A and B, $200 in C, or any similar combination). There are just too many possibilities and they are not known because the $1000 has become anonymous and notational in the e-Wallet after the fund-in.

As a result, all the merchants deny that the transactions made at their websites involve the dispute amount since the money was not traceable, and the e-Wallet provider lacks of evidence showing how the $1000 was divided up and spent.

At the end the e-Wallet provider is not able to solve the dispute with the merchants. Hence the provider suffers probably the charge-back amount and trust from both end-users and merchants.

As credit card payment via internet faces high fraud rate [3], whereby the payment solution provider cannot afford to take the risk in view of most revenue models are simply based on commission charged on the transaction amount, e-Wallet becomes less attractive a business model. This problem persists unless some innovative mechanism rolls out to safeguard the service provider's interest in accepting credit-card as a fund-in method.

3. Our Proposed Solution

One way to tackle the dispute issues on fund-in and transaction is to make the payment traceable. This however does not prevent fraud or stop culprits from making fraudulent transaction. Traceability means fraudulent transaction can be detected, picked up, stopped and most importantly, from the e-Wallet provider point of view, assertive evidence that can be shown is used to confront and blacklist the merchant(s) or users involved.

To achieve traceability of each payment transaction in e-Wallet system, the concept of Coupon [4] is introduced in the design of our payment system. Digital coupon in a sense is an indivisible monetary notation that has been adopted in implementing e-Cash [4][5] and as a ticketing or metering mechanism [6] for lightweight electronic commerce protocol [7] and online advertising [8][9]. In our payment system, the electronic form of coupon is not used for circulation over the internet, but as a traceable form of money notation for enabling fund-in transaction and the corresponding purchase transaction.

3.1. Coupon feature

Two balances are maintained at each e-Wallet account, namely Cash balance and Coupon balance. If the fund in is performed by bank transfer or debit card, the fund in will appear as Cash balance. If the fund in is performed

by credit card, then it will appear as Coupon balance in the e-wallet.

Unlike the coupon balance, the e-Wallet holder can fund in several times before sending out the cash balance. There is no restriction on how many times an e-Wallet holder can intake or send out fund from the Cash balance in his/her e-Wallet.

As for the Coupon balance, if the e-Wallet holder fund in a certain amount, say $100, the balance will look like a paper Coupon with face value of $100. This Coupon must be spent in whole by sending to a specified merchant who is also a registered e-Wallet holder of the system. The e-Wallet cannot fund in additional value, say $10 in addition to the first Coupon value of $100, before the previous Coupon balance is spent. In other words, each fund in transaction amount will immediately become a Coupon (token) stored in the wallet. At any one time, there would be either none or exactly one coupon in the e-Wallet. Each Coupon cannot be spent divisibly over multiple transactions, and each Coupon cannot be used for more than one merchant. It is not transferable too to another merchant. These rules are trying to ensure that every transaction that comes from a credit card becomes totally traceable. Its usage can checked too under this enforcement. The Coupon contains full details of the user, the source (credit card), and the destination (merchant).

Let x be the amount of fund-in
Let $X = [0..M]$ where M is the maximum amount allowed
Let $Y_i = <u, m, c, t>$ where i is the index of the Coupon and the variables in the tuple are defined as follow:

u = UserID, email address, login-name, password
m = MerchantID, email address, login-name, password
c = Credit Card number, Cardholder name, exp. Date
t = Time-stamp, IP address of the user

Figure 1 shows the operational details. With this design, the system offers a feature which allows traceability of the participated sender and receiver (the merchant) of each Coupon in a manageable manner. In addition, the design of the merchant wallet can be set to only receive Coupon and cannot send out Coupon. Should there be any fraud reported, the system operator can easily identify the "fraud Coupon" and know it has been sent to which merchant.

3.2 Fraud Manager System

With the Coupon feature, a Fraud Manager mechanism is implemented on top of the standard payment system for detecting possible frauds. The market does provide types of payment systems but none of them has a function to dedicatedly deal with security and trust issue associated with end-user behavior. Here we are not talking about network security whereby firewalls can be installed or 128-bits SSL is used. Suspicious user behavior leads to trails of fraudulent cases. A culprit user emulates a genuine credit card holder to make fraud transaction via the anonymous wallet system. Since the money representation in e-Wallet is usually anonymous if Coupon is not implemented, the payment system serves as a soft target to be abused for money laundering or other credit card crimes.

Having the Coupon strategy makes transaction traceable in the hope of deterring fraud attempts. Furthermore, we have to learn the pattern of culprit people behavior so that fault logic can be developed to disallow their attempted transaction. The objective is not to identify the bad guy but identify possible fraud transaction and let it down. That is, we learn the trails of fraudulent transactions and we preemptively stop a transaction that is similar to those fraud cases.

Most commonly standard measure taken by bank payment gateway is to restrict the credit limit, monitor the frequency of access per day/week/month, etc., and create a black list to block the suspect transactions.

These are not enough as culprit users will try the allowable credit limit and under the maximum frequency, and use cards that are not yet black-listed where possible.

There is no once-off solution to deal with those fraud attempts. The approach is to keep on monitoring the suspect behavior and develop fault logic to block the suspect transaction. The term fault logic we used here is a set of rules for dealing with detecting a potential fraud case. The rules are derived and "learnt" over time from past fraud history. For a start, these rules are manually defined by human experts who are familiar with credit-card crimes and our payment system. In a later stage, machine learning algorithms such as data-mining [10][11] and neural network [12] can be applied to fine tune the fault logics. Therefore, the fault logic module must be designed to be extendable as new measure (rule) is added in to minimize fraud every now and then.

3.3 Fault Logic Module

Firstly, rule-based logics that were called fault logic have to be developed. In the initial phase, the following functions are programmed in the fault logic module.

(1) Smart black list – usually we cannot decide which credit card to be suspected as fraud and put into the black list database without a transaction history. The most common way is that when there is a transaction already happened and reported as fraud, the corresponding transaction will be identified fraudulent. The Coupon feature has ensured each fund-in transaction into the e-Wallet (at that moment the Coupon has not been spent) is

stored as a whole amount and completely traceable. So any credit card that was reported as fraud, the corresponding Coupons generated by that credit card can be seized, the intended merchant and the user of the e-Wallet would be known. Even though the Coupon has sent out, we know exactly to whom it was sent and the amount of it.

Here the function of smart black list is that other than simply creating a black list database for administrator to input credit card numbers for the fraud manager to block the transaction, there is a "linkage black list table" (Suspect List) built. We define the linkage by detecting if any already black-listed card has ever been used at an IP address and other cards being using at the same IP address, that card will automatically be entered into the table.

If any e-Wallet transaction by email format has been performed with a black list card, any other card (though not in black list at the moment) being used by the same e-Wallet will be entered into the table.

Further to above rule, by tracing other suspect cards (that have not been black-listed yet), other e-Wallet(s) that have links to will be identified. Then additional cards ever being used by these e-wallet(s) will be entered into the table. This is based on the assumption that a user may use several cards across several accounts. When one of these cards or accounts is detected fraudulent, all the other related ones will be suspected.

Technically, once a transaction is detected to be a fraud, all the parameters associating with that transaction would be highlighted and put into the Suspect list. The parameters in the suspect list will be used to trace out other e-wallets and cards that have not yet been declared as fraud.

The rule though sounds a bit complicated. The principal is that if there is any linkage (originating from a black-listed card) to any e-wallet and subsequently any (not yet black listed) card will all be traced out. Upon tracing out these will be automatically entered to the "linkage black list table" which forms the smart black list. This is a step up measure to preemptively block a transaction performed by a particular credit card, before being formally reported by the bank that is a fraud transaction by then it is too late.

(2) IP blocking – if any credit card prefix indicating the issuing country when being used compared to the country location of the IP address for each real time transaction is not the same, the transaction will be blocked. This is optional as a user who holds a card issued from a country actually use it from another country.

(3) Different public domain blocking – if the e-Wallet is registered by using a public domain email address, e.g. apple@yahoo.com, apple@hotmail.com, apple@sohu.com, then all these transactions will be blocked. Once again, this rule is optional. But according to our experiences, almost in all the fraud cases public email addresses are used. This avoids being traced by ISP.

(4) Different receiver e-wallet blocking – if one single sender e-wallet attempts to make x transactions with y merchants, i.e. transfers funds to different merchant e-wallet(s) within z timeframes, the transaction will be blocked. Of course the variables x, y, z mentioned above can be changed by administrator.

The above-mentioned "IF-THEN" rules are developed after researching the behavior of many reported fraud transactions. This is not a one-time exercise and continuous efforts will be made to cultivate the latest patterns of fraud behavior hence new fraud logics will be added on from time to time.

From the view of implementation, the *FaultManger* class is actually an RMI server [13] that is event-driven and does real-time processing. A *RMIRemoteObject* is bind to this server such that any payment system can be connected to the *FaultManger* server that makes it integration flexible. The technical team who are responsible for the payment system only needs to know the structure and defined data of the *RMIRemoteInterface* and they could develop simple client to connect to the fault manager.

FaultLogic is an interface such that any class that extends it will inherit the method *doVerification*(). By using the *FaultLogic* interface, new checking rules can be added to the system without affecting the other parts of the systems. New checking rules can be loaded dynamically into the system by using the Class that comes with Java API. The design of the class *FaultManager* in UML notation and its operation can be found in Figure 2 and Figure 3 respectively.

4. System Design

This internet-based system will be designed as a platform that facilitates multi-channel and multi-purpose e-Wallet system, with most standard payment functions plus the fund-in channel of credit card payment. The standard functions include internet payment, account transfer, account enquiry, customer service console enquiry, web-based system administration, fund-in and fund-out transactions. The operation however is depicted in the form of data-flow-diagrams for both merchant and client in Figure 4 and Figure 5 respectively.

Some specified group of e-Wallets are entitled to the loyalty program. The administrator can configure the detail of each loyalty program simply by monitoring the amount of transactions incurred everyday. A web based back-end system administration console provides the functions of card information enquiry, card administration, report download, user administration, and system maintenance.

4.1 Cash flow

If the e-wallet holder wishes to fund in by bank transfer, he has to login his account entering the details of bank transfer. The system will then automatically generate a bank-in request. The operator then approves or disapproves the bank in request. If approved, the e-Wallet cash balance will be increased.

Alternatively the e-Wallet can be funded-in by credit card via an interface with bank's payment gateway. Then in the e-Wallet a Coupon balance will be generated. It should be noted that the next fund in by credit card would not be accepted by the system until the current Coupon balance has been cleared from the e-Wallet. During the fund-in the e-Wallet holder will be directed to the payment gateway in real time to complete the fund in process. After the payment gateway has approved the transaction, the e-Wallet holder will then be re-directed back to the system and his e-Wallet will hold a Coupon balance.

Transfer between e-Wallets can be performed easily via the web front-end. This facilitates a buyer sending a payment to a merchant. The e-Wallet holder simply login to his e-Wallet and selects the "send fund" function. The system will prompt to input the receiver wallet ID and after successful authentication, the value is transferred from the sender's e-Wallet to the receiver's e-Wallet. This is similar to PayPal except one thing: The Cash balance can be sent out partially across several transactions, but the Couple balance must be sent out in a whole sum.

Other than transfer fund between e-Wallets, this system also allows e-Wallet holder to settle payment using his e-Wallet. E-wallet holder selects products/services into the shopping basket at the authorized integrated merchant web site, and chooses to pay by e-Wallet. He will then be URL redirected to the electronic payment system. The payment information including transaction amount, merchant transaction reference, product code, currency, and merchant ID, merchant's URL for successful transaction will also be carried forward in the redirect link. In any case, the e-Wallet holder is prompted to enter the wallet ID and PIN. The system performs user authentication and checks e-Wallet balance for the payment.

After making the payment the e-Wallet holder will be URL redirected to merchant's URL for both successful and unsuccessful transactions. The information including merchant transaction reference, system transaction reference and return status that indicates whether the transaction is successful or not will be forwarded to the redirect link.

4.2 System module interoperability

Conceptually, the payment system will be inter-connected with internet environment, merchant web site that supposed to integrate with the payment system, payment gateway which facilitates the payment information transfer and real time payment. Additionally, the system offers interoperability with telecom switches such as SMS gateway, and WAP-enabled applications.

4.3 E-Wallet system

The e-wallet system is where the core logics for balance storage, transaction history of the e-wallet, and the overall management of card accounts are implemented. It stores e-Wallet holder account information, such as card balance, PIN, expiry date and card status. A bank is required to provide a payment gateway interface for the credit card fund-in and payment. Written in Java, a communication interface is always provided for the merchant site who wants to easily integrate with our payment system. The front-end server allows cardholders to access their e-Wallets through an internet browser. For security reason, sessions of SSL are provoked between the browser and the front-end server.

4.4 Commission Feature

A commission feature is designed for the system owner to earn from the electronic payment service provision. Depending on the transaction type, a variable commission may be charged on each transaction. The commission parameter is generally defined using this formula: Commission = (Transaction amount * commission rate %) + a fixed charge.

A commission plan defines the corresponding values in the above formula for different transaction types. Each e-Wallet can be associated with a customized commission plan.

Internally, there is a commission e-Wallet owned by the system administrator for each available currency defined in the system to collect the commission amounts.

When making a transaction, e-Wallet holder enters the transaction amount and the above formula will be used to calculate the commission transparently. The commission amount will be charged to the e-Wallet at the same time of the transaction. If the e-Wallet balance is insufficient to cover both the transaction amount and commission amount, the transaction will be rejected.

4.5 Account Class Feature

Four account classes are supported, namely Personal, Premium, Business and Merchant. Each e-Wallet account

is instantiated in Java programming sense, from one of these four account classes and thereby inheriting the common data and functions.

Different risk rules may apply for different account classes. When a user signs up an account, the default "Personal" and/or "Business" class may apply. Upgrade to "Premium" and/or "Merchant" will be done manually through the backend console. Only Merchant accounts are allowed to receive Coupon. Only Personal or Premium accounts are allowed to fund in and send Coupon.

5. Conclusion

The implementation of our proposed e-payment system is specially designed with a Fraud Manager sub-system which emphasizes on preventive measures to possibly minimize fraud. We admit that in no circumstance fraud can be 100% eliminated. However, most current payment systems have not been designed in a way to tackle this issue. Fraud transaction is a real problem that hampers the growth of e-commerce and must be overcome. According to our experiences, a good way to wining merchants and users in using our e-Wallet system relies on how well fraud and dispute can be handled by our system. In this paper, we have discussed the Coupon concept and its implementation issues for a payment system that is capable of detecting frauds.

6. References

[1] Jon M. Peha, Ildar M. Khamitov, "PayCash: a secure efficient Internet payment system", Proceedings of the 5th international conference on Electronic commerce, Pittsburgh, Pennsylvania, pp.125-130, September 2003.

[2] J. D. Tygar, "Atomicity in electronic commerce", Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing, Philadelphia, Pennsylvania, United States, pp.8-26, 1996

[3] National Fraud Information Center: fraud.org or ftc.gov

[4] Patiwat Panurach, "Money in electronic commerce: digital cash, electronic fund transfer, and Ecash", Communications of the ACM, Volume 39, Issue 6, pp.45-50, 1996

[5] Levy, S. E-Money (that's what I want). Wired 2, 12 (Dec. 1995) http://www.hotwired.com/wired/2.12/features/emoney.html

[6] R. Anand, M. Kumar, and A. Jhingran, "Distributing e-coupons on the Internet", In 9th Conference on Internet Society (INET '99), San Jose, 1999.

[7] M. Franklin and D. Malkhi, "Auditable metering with lightweight security", In R. Hirschfeld, editor, Financial Cryptography (FC '97), volume 1318 of Lecture Notes in Computer Science, pages 151-160. Springer-Verlag, Berlin, 1997.

[8] M. Jakobsson, P. D. MacKenzie, and J. P. Stern, " Secure and lightweight advertising on the web" In 9th World Wide Web Conference (WWW9), 1999.

[9] Carlo Blundo, Stelvio Cimato, Annalisa De Bonis, " Advertising and Security for E-Commerence: A lightweight protocol for the generation and distribution of secure e-coupons", Proceedings of the eleventh international conference on World Wide Web, May 2002, Pages: 542 - 552, Honolulu, Hawaii, USA.

[10] F. Bonchi, F. Giannotti, G. Mainetto, D. Pedreschi, "A classification-based methodology for planning audit strategies in fraud detection", Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp.175-184, San Diego, California, United States, August 1999.

[11] Chan, P.K.; Fan, W.; Prodromidis, A.L.; Stolfo, S. J, "Distributed data mining in credit card fraud detection", Intelligent Systems, IEEE [see also IEEE Expert], Volume: 14 Issue: 6, pp.67 -74, Nov.-Dec. 1999

[12] Ghosh, S.; Reilly, D.L, "Credit card fraud detection with a neural-network", IEEE System Sciences, 1994. Vol.III: Information Systems: Decision Support and Knowledge-Based Systems, Proceedings of the Twenty-Seventh Hawaii International Conference on , Volume: 3, pp.621-630, 4-7 Jan. 1994

[13] Wall, T.; Cahill, V, "Mobile RMI: supporting remote access to Java server objects on mobile hosts", IEEE 3rd International Symposium on Distributed Objects and Applications, pp.41-51, 17-20 Sept. 2001

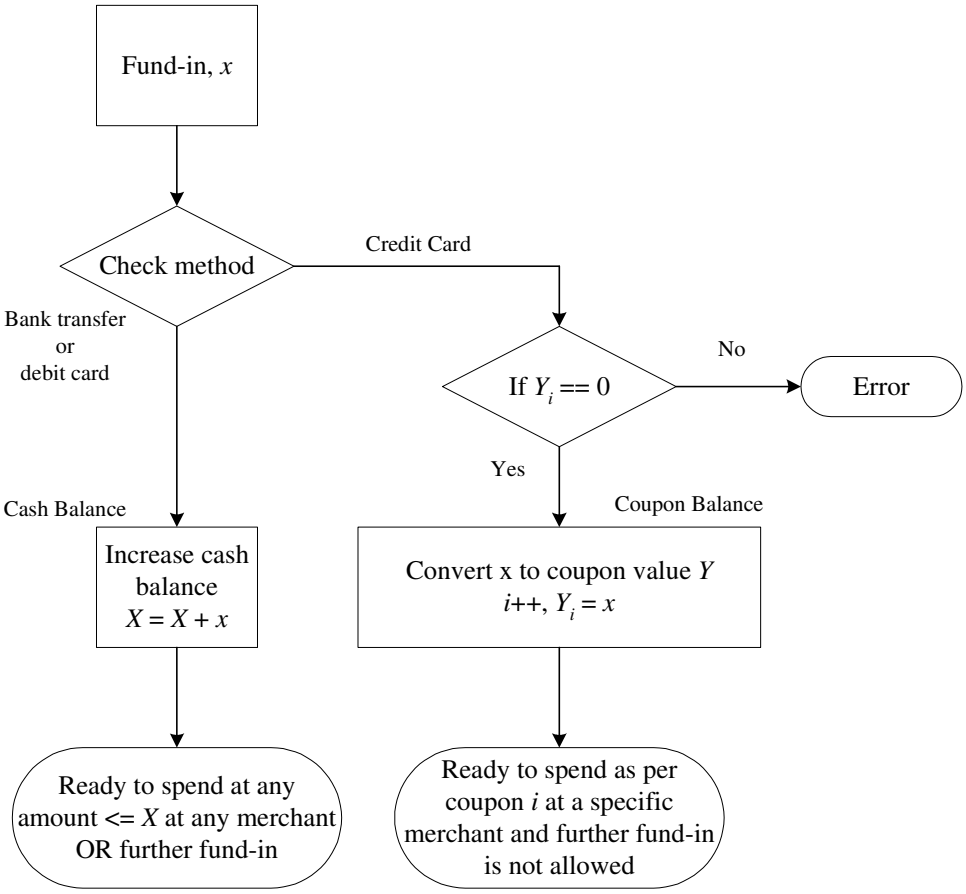

Figure 1. Flow-chart of the fund-in logic for Cash Balance and Coupon Balance inside the e-Wallet account

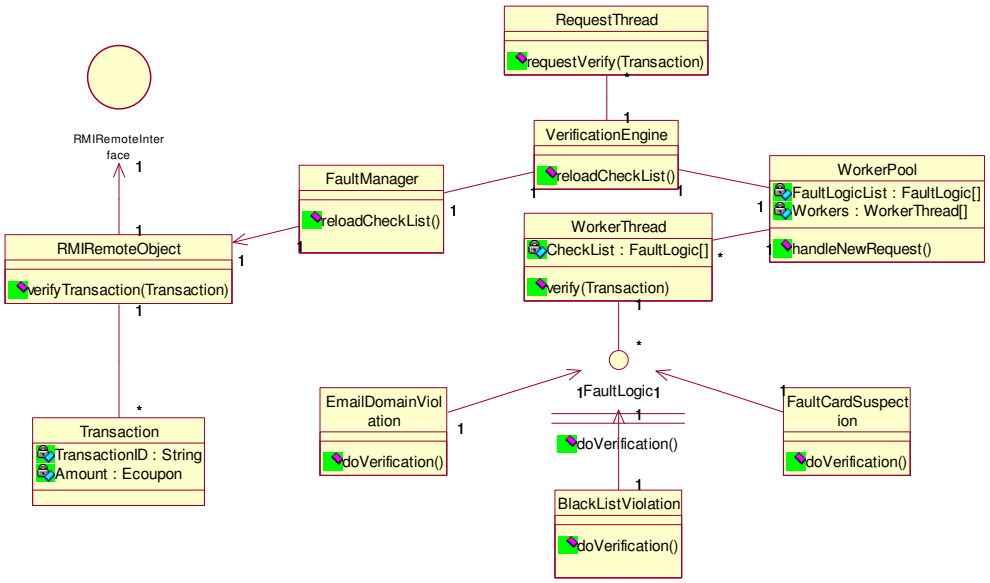

Figure 2. Class Design of Fault Manager in UML

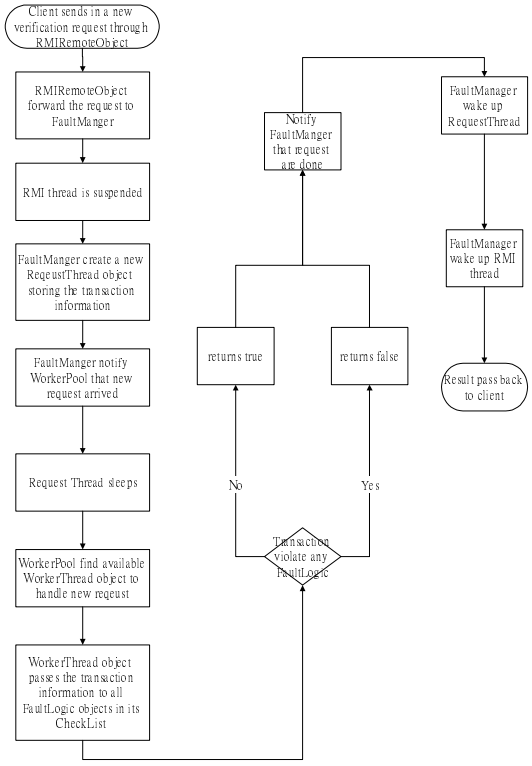

Figure 3. Work flow that demonstrate how Fault Manager works

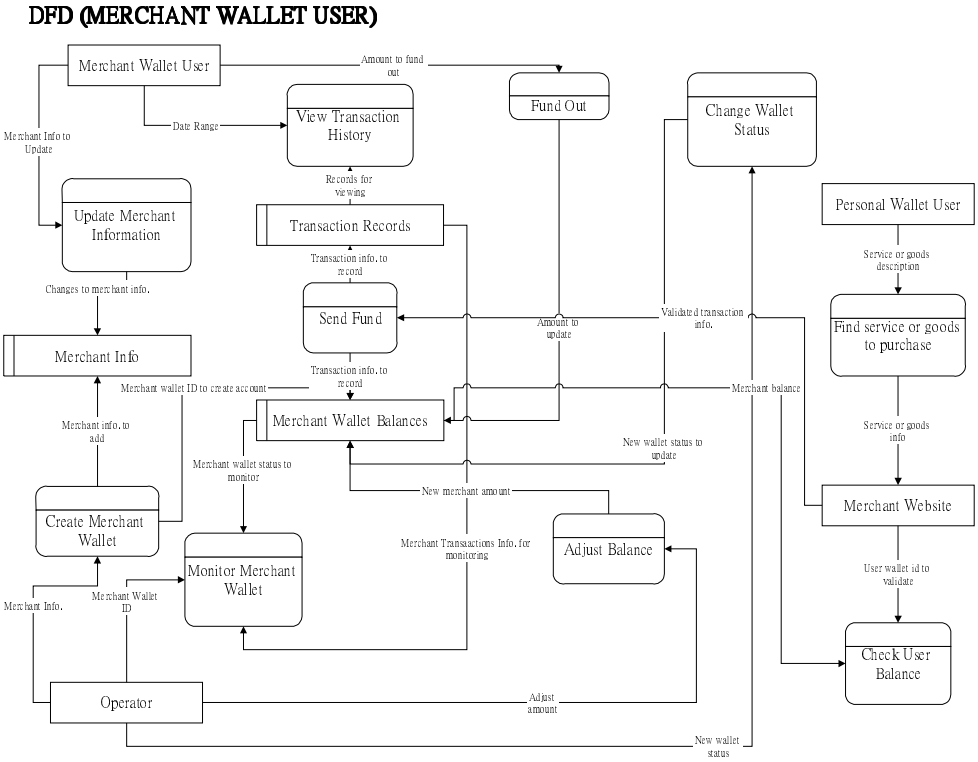

Figure 4. DFD diagram of e-Wallet account used by Merchant

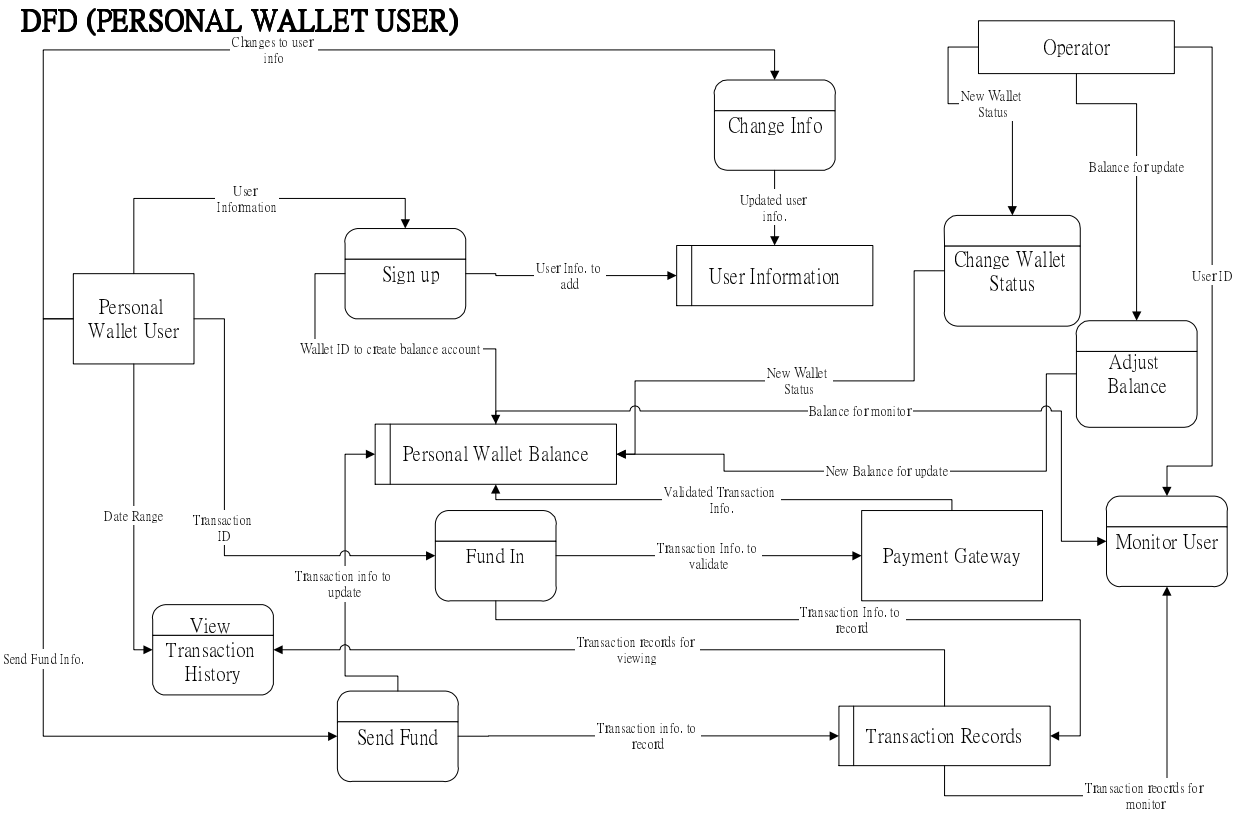

Figure 5. DFD diagram of e-Wallet account used by personal user