CyberTrade 2 Payment Server and CyberBank

Project Documentation

Arrianto Mukti Wibowo
iscp9063@nus.edu.sg

School of Computing
NATIONAL UNIVERSITY OF SINGAPORE
Kent Ridge Crescent, Singapore

February, 2000

1 Introduction

CyberTrade 2 is a centralized payment server for use in CS4260 Electronic Commerce course at School of Computing, National University of Singapore. It is used as a support tool in the project assignment in CS4260. It helps the groups by freeing them to develop their own payment system & cashier.

Since CS4260 also has a 'trading session', where the students shop at other groups' website, CyberTrade 2 also increases security. Each student is given a 'credit card number' to be used in the payment process. Since the payment process is done at CyberTrade 2 payment server, not at the merchant website, the merchant will not be able to know the credit card information of the customers.

The students are able to see how much money they have spent on shopping, at CyberBank.

Notes are given to describe certain aspects which is simulated in this project. Likely real world implementation is also described to give the readers more understanding.

Section 2 describes the requirement analysis of CyberTrade 2. Section 3 elaborates more on its design specification. Section 4 discuss CyberTrade 2 implementation in CS4260. Section 5 discuss the possible future development of CyberTrade 2. The appendix shows how a merchant can easily connect to the payment server.

2 Requirement Analysis

CyberTrade 2 Payment Server was designed to meet the requirement of CS4260 simulated e-commerce environment to let the students trade among themselves. Among those requirement are:

1. There exist a mechanism to let only the students trade among themselves
2. The students are given a certain amount of money to purchase merchandizes.
3. There exist a mechanism to analyze the shopping pattern of the students.
4. There exist a centralized payment mechanism, so the 'merchants' (groups) do not need to develop their own payment mechanism.

Here we list the brief system requirement for CyberTrade 2:

1. Centralized payment server: a single payment server for all merchants, as the gateway to the financial institution.
2. Pseudo-anonymity: The merchant do not need to know the true identity of the customer.
3. Minimize credit card information scatter: in other words, not letting the merchant to have customer's credit card information.
4. Standard HTTP FORM passing: merchants can use whatever tools they want to develop their website, since the the payment server uses standard HTML passing to pass parameters.
5. Only registered merchant can use the payment system.

Since CyberTrade 2 is a course support tool, the requirement is not stringent, especially the security aspect. We leave the security aspect for future development.

For the CyberBank, the requirement is quite simple:

1. Cardholders (students) can check how much money they have spent
2. Cardholders can see a report, where did they spend their money
3. Only truthfull cardholders are allowed to use CyberBank.

3 Design Specification

3.1 Solution to the requirement

The basic design of CyberTrade 2 is the three-party payment system scheme. Instead the merchant connects directly to the financial institution, the merchant connects to a service provider, which provides the payment gateway to the financial institution. Consequently, each merchant does not need to develop or maintain their own connection to the financial institution to clear the payments.

Transaction between customer's browser, merchant's webserver and CyberTrade's payment server (CTPS) is done through the Internet. Authorization process from CTPS uses the existing credit card network infrastructure.

Since only the students with credit card number can be a customer in this simulated e-commerce environment, it meets our requirements. Each customer should check at CyberBank to verify how much money they have spent with their credit card. Upon entering the CyberBank, the cardholders are asked to type-in their credit card numbers.

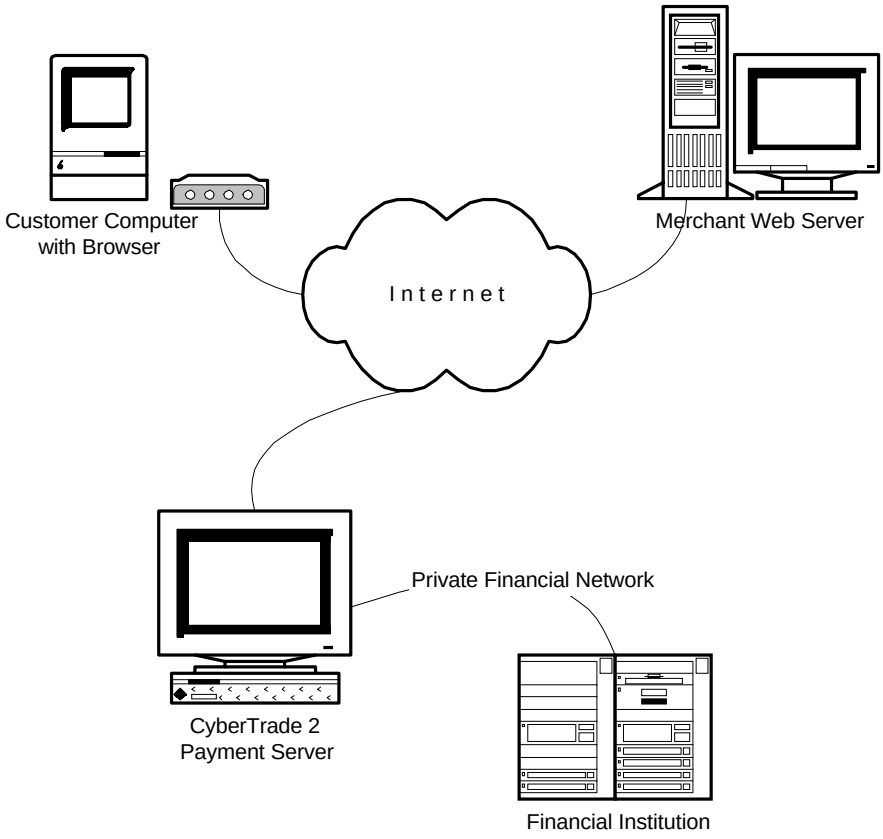

Figure 1 Three party payment system

3.2 Detailed transaction protocol of CyberTrade 2 Payment Server

1. Customer browses merchant's website. Customer selects the merchandizes he wants to buy and puts them into the shopping cart.
2. At merchant's check-out point (cashier), the merchant consolidates customer's shopping list, plus delivery cost if necessary. Then the merchant generates a purchase slip and sends the purchase slip to the customer.
3. The customer verifies the consolidated total amount to be paid. If the customer approves, the merchant redirects the payment to CyberTrade Payment Server (CTPS).

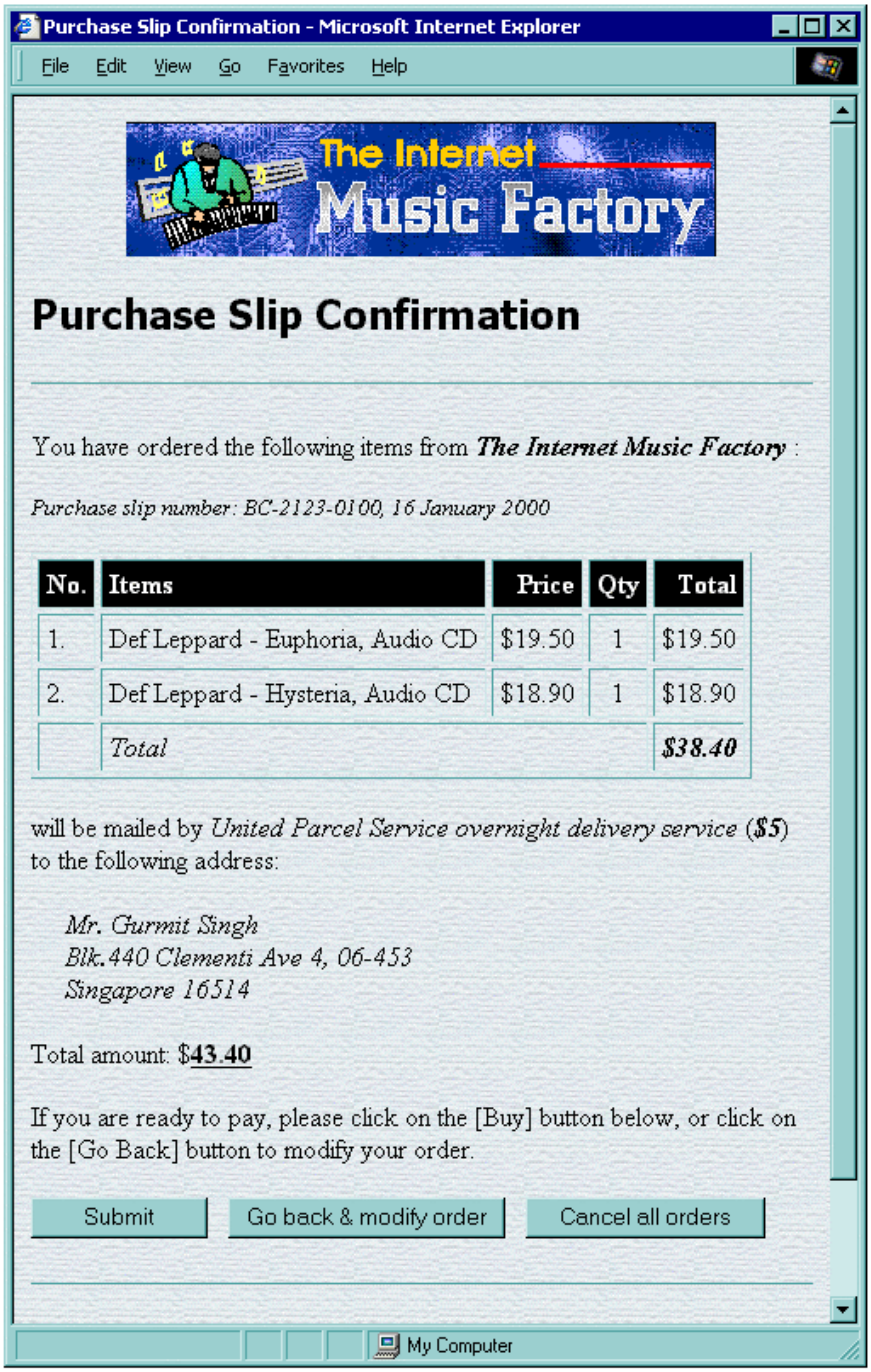

Figure 2 Purchase slip at the merchant

4. CTPS receives the approved purchase slip and generates a payment slip.
5. The customer fills the payment information such as his credit card number, expiry date, and other necessary information.

Figure 3 Payment Slip at CyberTrade 2

6. At this point, the payment authorization process is done through a standard credit card payment authentication.
7. Note that there are various ways to conduct credit card authorization, in the real world. In a simple real world CTPS implementation, CTPS does not authorizes the payment information on-line, but batches the completed payment slip to be cleared later to the aquirer. Since this project is in a simulated environment, the authentication is simply done against a set of already issued credit card database.
8. Upon receiving the authorization result, CTPS forwards the authorization result to the customer.

Figure 4 Authorization Result generated by CyberTrade 2

9. The customer receives the authorization result, and the customer is redirected to the merchant along with the authorization result.
10. The merchant receives the authorization result, and prepares for the delivery.

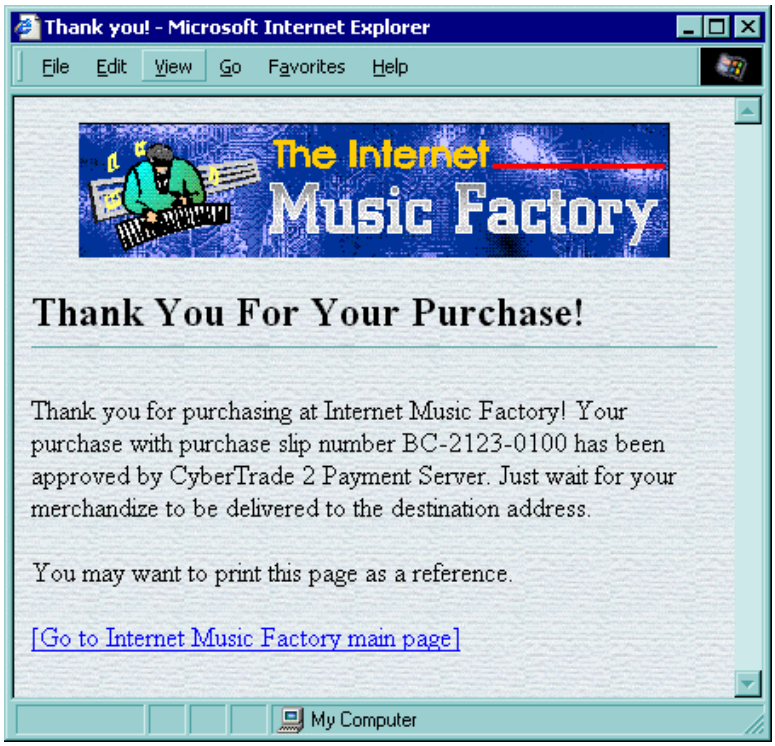

Figure 5 Back to the merchant after authorization

11. The customer receives delivery notification and merchant delivers the merchandize.

3.3 Message sequence chart of the transaction

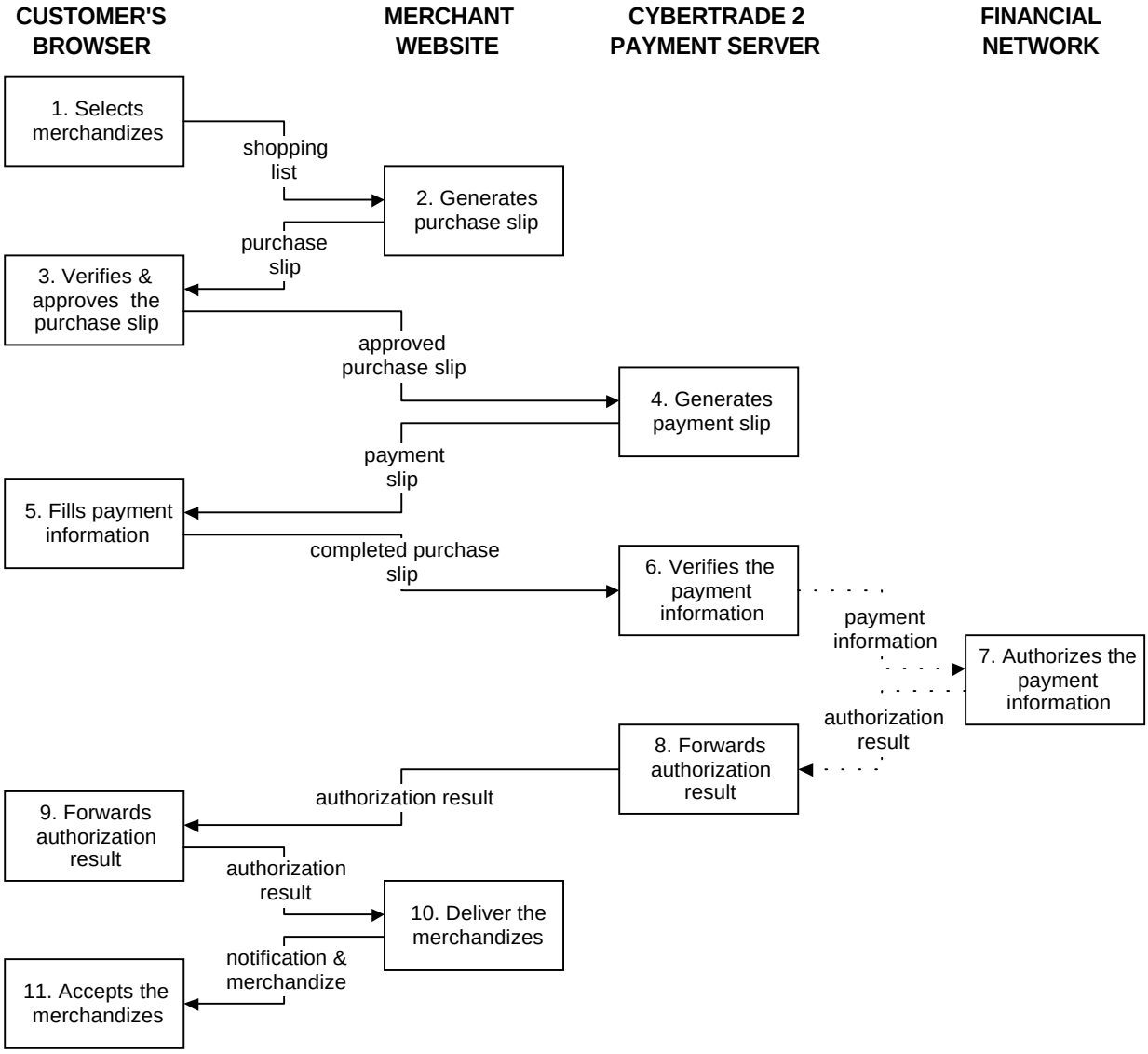

Figure 6 Message sequence chart of CyberTrade 2 transaction

3.4 CyberBank

There are only two pages at CyberBank. The first screen is the login page, and the second page is the report of transaction history.

3.5 Data structures & processing

3.5.1 PurchaseSlip

The following data must be passed from the merchant to the payment gateway at the time the customer approves the purchase slip:

1. PurchaseSlip: a serial number generated automatically by the merchant for each purchase slip.
2. MerchantDate: merchant date stamp.
3. MerchantTime: merchant time stamp.
4. MerchantID: merchant unique identification.
5. PurchaseInfo: the merchant should give some brief description about the purchase, just to remind the customer when the customer recieves his monthly credit card bill.
6. ReturnURL: a URL where CyberTrade must point after authorization process.
7. MethodReturnURL: because the ReturnURL is executed as part of a form processing from CyberTrade 2 payment server, the customer must decide how CTPS may pass necessary parameters (including AuthorizationResult) to his ReturnURL. Two methods are possible: GET and POST.
8. Amount: the total amount of the payment

The merchant do not need to pass all the payment detail such as items purchased to CTPS. The merchant must also calculate the delivery/handling cost before submitting the total amount to the payment gatway.

The merchant should generate the purchase slip on the fly. Note that the merchant do not need to display all of the information to the user. The merchant must save the purchase information, for further processing. Later after receiving the authorization result from CTPS, the merchant needs to recall this purchase slip record.

3.5.2 PaymentSlip

CTPS will receive all of the paramater from the purchase slip and use them to generate the payment slip. Additional information in the payment slip include:

1. CardholderName
2. CreditCardNumber
3. CardBrand
4. ExpiryDate
5. PaymentSlip: serial number of the payment slip, generated automatically by CTPS.
6. ServerDate: date stamp on CTPS.
7. ServerTime: time stamp on CTPS

CTPS uses the credit card information (from 1 to 4) to verify the credit card information. However, in the current version, CTPS only checks the CreditCardNumber.

When CTPS displays the payment slip, it will substitute the MechantID with MerchantName. Therefore, the customer will see the merchant name on the payment slip, not the MerchantID. Observe that the payer does not has to be the same as the reciever of the merchandize (addressee).

3.5.3 AuhtorizationResult

After CTPS verifies and authorizes the credit card information, it generates the authorization result, which includes the following:

1. ReportAuthorizationResult: there are two possible outcome, which are ‘authorized’ and ‘unauthorized’.
2. PurchaseSlip: the associated purchase slip serial number.

The merchant must process those information according the HTTP parameter passing method they choose before (GET or POST method). If the authorization result for the associated purchase slip is

Note that in the real world sitation, CTPS may check whether the credit card number validity, just by examining the credit card information, with some (supposed to be secret) algorithm. But for authorization, CTPS needs to connect to the financial institution. For example, CTPS needs to know whether the credit ceiling has not been reached, or whether the card is revoked or not. CTPS can do the authorization result by bathcing all of the payment slip information at the end of the day to the financial institution. Therefore, in the real world, the merchant will likely get the final authorization result by some sort of notification other than HTTP, such as a authorization report via e-mail each day. However, in our project, the authorization is simplified only at CTPS.

4 Implementation

This section discuss the implementation issues of CyberTrade 2 in CS4260, for the semester commencing January 2000.

4.1 Software tools

Among several tools assessed during the preliminary research, MS Visual Interdev 6.0 was choosen to be the primary development tool. Other candidates was Perl, Java familiy tools, and Oracle Web Server/Development Tool. The reasons are:

1. MS Visual Interdev 6.0 allows rapid development of websites, with complete suit of management tools. It also separates program with web design, which helps a lot in website development.
2. NUS has licenses on MS Visual Interdev 6.0.
3. Easy database integration using ODBC.
4. Better web server memory management (compared to Perl).

We use MS-SQL Server 7.0 as the database server for CyberTrade 2 Payment Server. It also doubles as the merchant database server. The reason to use MS-SQL Server 7.0, instad of Oracle Server is because the only license which is currently free is MS SQL Server 7.0.

CyberBank uses the same tools also.

4.2 Implementation of the scripts

CTPS and CyberBank program scripts are Active Server Pages (ASP). ASP is actually a server side VBScript. Since the web sever processes the ASP script before the resulting HTML is passed to the client browser, any kind of browser can display ASP pages.

In the current implementation, only the first 8 digit of the credit card number is verified by CTPS. The 8 digit is split into two fields (each 4 digits), which are CCNumber1 (the first 4 digit of the credit card number) and CCNumber2 (the next

4 digit). In our implementation, the CCNumber1 acts like the primary key for the cardholder, and CCNumber2 acts more like the 'password'. Note that in real Internet credit card transaction, mostly do not involve 'passwords'.

5 Further Development

1. The most obvious future development path of CyberTrade 2, is improving its security. Public-key encryption on the merchant side and CTPS will significantly improve the security factors (privacy, authenticity, integrity and non-repudiation). Using SSL as a security measurement of CyberTrade 2 is not sufficient, because the client's browser points to several locations during payment transaction. Merchant and CTPS-side specific cryptographic functions/modules must be implemented.
2. The shopping cart may also be integrated into CyberTrade 2. Therefore its role can double not just as a payment server, but also a delivery service provider. In this case, CyberTrade will need to connect to several delivery service such as UPS, Fedex, or DHL.

Appendix A: Connecting to CyberTrade 2

5.1 Creating the PurchaseSlip.html (or .asp, .htm, .*)

These are the required parameter in merchant's purchase slip to be passed to the CyberTrade 2 Payment Server. Note that the merchant must save these information to its own database server (for management purposes, such as delivery), prior sending purchas slip information to the payment server.

Example:

```
<FORM action=http://socmgl129/CyberTrade2/PaymentSlip.asp method=post name=FORM1>
<!--Here is the hidden information to be passed to PaymentSlip.asp at the payment gateway -->
    <INPUT name=PurchaseSlip type=hidden value=BC-2123-0100>
    <INPUT name=MerchantDate type=hidden value="16 January 2000">
    <INPUT name=MerchantTime type=hidden value=19:00>
    <INPUT name=MerchantID type=hidden value=group1>
    <INPUT name=PurchaseInfo type=hidden value="Internet Music Factory on-line shopping
    (UPS ovn)">
    <INPUT name=ReturnURL type=hidden value= http://socmgl129/group1/
    BackToMerchant.asp>
    <INPUT name=MethodReturnURL type=hidden value=post>
    <INPUT name=Amount type=hidden value=43.40>
<INPUT name=submit1 type=submit value=Submit>
</FORM>
```

5.2 Creating a page to respond the Authorization Result

Our example, uses the file named BackToMerchant.asp, which is an ASP script. However, you can use any web programming tools you want. Note that the variable passing method depends on how the merchant set the **MethodReturnURL**.

```
<%@ Language=VBScript %>
<%
    thisReportAutorizationResult=Request.Form("ReportAutorizationResult")
    thisPurchaseSlip=Request.Form("PurchaseSlip")
%>
<HTML>
```

```
<BODY>
<p>The authorization result for purchase slip number <% =thisPurchaseSlip %>  is :
<%=thisReportAutorizationResult%></p>
</BODY>
</HTML>
```

Note: possible value for **ReportAutorizationResult** is ‘authorized’ or ‘unauthorized’.