

COMP3334 Project Report

CHENG Mingrui

LIAO Zhengyun

PU Yuzhou

WU Yuanlin

1. Background

In technological perspective, electric commerce is defined as a commerce style which is capable of providing to customers a service of purchasing and selling products on the internet or other online platforms. With the development of technology, electronic commerce has enjoyed a high level of prosperity and become one of the most successful financial business forms (Hsieh, 2001). The e-payment method, which is the basis of e-commerce, has also experienced a fast growth and has become one of the most important payment methods nowadays. It is reported that in Korea alone, the market size of e-commerce has reached 507.42 USD and has grown by 34.6% compared with the previous year (Kim et. al., 2010).

Though being popular and common nowadays, the concept of e-commerce was once a scientific and unimaginable concept when public barely had idea about the internet and the cyber space was an obscure concept (Kalakota& Whinston, 1997). There were two important changing points in the history of e-commerce. One is the development of personal computer in 1983, which enables people to have their personal electronic devices that could enable them to depend daily tasks partly on electronic devices, though at that time a personal computer was still heavy and dumb and was not portable; another was the establishment of local network in 1989, with which the cyber space could be realized and concepts once could only see in science fiction books had come into reality.

The influence of e-commerce to traditional business model is enormous. Its impact even become greater than many inventions like television in 1950s or video players in 1980s. It has cast tremendous influence on people's life, first from the internal change of firm operation mode, and then further influence external operation of other institutions like the network of suppliers and government agency.

Meanwhile, as an important foundation of supporting the electronic commerce, electronic payment has aroused tremendous attention. This new way of payment is no way like any of its previous payment methods, introducing completely new concept to the public. However, while seeing its advantages over traditional payment methods, its risks is also being concerned. Payment security is the most critical concern of e-payment system, as once an attacker can find a vulnerability in the payment system, the property loss could be tremendous. Once the accident happens, not only will electrical commerce suffer great setback, the public's faith towards public payment platform would be also greatly challenged.

Thus, understanding the importance of security issue towards electronic payment system, this project is going to implement an online payment platform with the consider of various aspect of security issue. An pay-pal like online payment platform is established, with users could register for an account, add their payment method (which is credit card number in this project and more kinds of payment method could be added later according to actual use case), request or send money to another user in the website, view personal payment and activity history, or pay for a product in another website through the payment link to this payment

website.

The framework used for this project is node.js, with html, JavaScript and CSS to be the front-end language. This article will go through security design framework, actual implementation of security method and the basic workflow of the website, attaching three use cases.

2. Security Requirement Analysis

(1) CIA Triangle

1. Confidentiality

Confidentiality refers to the concealment of some sensitive information as well as data resources. It could also be interpreted as prohibiting unauthorized third party from accessing information that is sensitive or important. This need of keeping information secret is critical for some government resources as well as personal information, as the result of leaking sensitive information could be as serious as identity fraud, resulting in legal as well as security problems.

A breach of confidentiality means that an unauthorized party has gained illegal access to information that should not be revealed to them. To avoid breach and maintain information confidentiality, the authorize process of the website should be strict and precise. Apart from validating user's identity, additional efforts should be spent to ensure the data source security, like encrypt sensitive files, manage data access by hiding sensitive information to certain group of users, physically secure devices by storing them in the form of printed material in an offline environment and update security method or firewall.

It is also important to manage the usage of data. Only access sensitive information when it is necessary could help to prevent unnecessary information leakage.

In this case of establishing a payment website, confidentiality could be further specified to not leaking user's personal information such as email, phone number, credit card number and so on. More importantly, user's password should be protected and encrypted so as to keep the user account safe.

2. Integrity

Integrity refers to the consistency, accuracy and authenticity of the data for the entire period of the data's life cycle. It is to confirm that the information is not altered, whether maliciously or accidentally at transit.

To maintain data integrity in the online payment platform, additional information should be added to the data resource to verify and validate data, like the checksums value. This is to prevent any non-human changes to the data, like server failure or some accidental change to the resource when transferring resource through network.

Besides, the authority of accessing data should also be controlled. This is to prevent data from being changed by some unauthorized people and to maintain data integrity. For example, nobody should be able to change other people's payment history in the payment system; and that the payment request should not be altered when transferring through the internet.

Data backup and version control should also be applied to the website in later maintenance.

This could help to find back all essential works after an accidental corruption of data, like the website files and frameworks.

3. Availability

Availability refers to the immediate access to the data as well as the reliability of data source. This could be threat by DOS attack or server instability.

To ensure the data availability, the e-payment website could adopt data backup resource and give hardware timely maintenance and keep with pace with the system update in order not to meet any problem because of the old version of the server or libraries used in the program. The website could also adopt disaster recovery plan. Additional security methods such as firewalls and proxy servers could be used to avoid the system from being unavailable because of malicious attack.

(2) Possible Threats

a. Denial of Service

The DoS attack refers to a kind of cyber-attack where the attacker intends to make the web server or network inaccessible (McDowell, 2004). The methods to launch the DoS attack including flooding numerous request to a certain server to occupies its resource, and thus resulting in the server is unable to serve other normal users and the service is down.

This attack would result in a delay or denial of service to other users of the website. If the payment platform is down, transactions would not be able to process and may results in serious problem. To prevent this, special actions are made to prevent DoS attack, which would be explained in details later.

b. Snooping

Also known as “sniffing”, snooping is a process of listening to the public network traffic IGMP snooping could be used to launch DoS attack when exploited, and other kind of snooping may also results in information leak and should be prevented(Xiaohu, 2009).

As the platform we are constructing is a payment platform, personal information leak or password leak is a critical issue. Thus, https is used to prevent snooping, which will be explained in details in later session of the report.

c. Usurpation

This refers to the unauthorized control of the system. This may lead to violation of data confidentiality and integrity. For example, one user should not be able to change another user’s personal information or payment history, and that unlogged in user should not be able to access user home page. To prevent unauthorized changes to sensitive information of the website, user’s identity is examined every time he visits a website that requires logged in identity. If he was not in the logged in stage or his log in information has expired, system would be redirected to the sign in page.

Security Design Specification

(1) SSL/HTTPS



SSL stands for Secure Sockets Layer, it is the standard technology for keeping the internet connection between two parties secured and encrypt the data transferred between the systems, also preventing the data being attacked and changed. We use a free approach to get a SSL certificate and secure our online payment system.

The SSL certification is acquired from Let's Encrypt website, which is supported by Google Chrome, Mozilla Firefox and other web browsers. Here we list how security requirements can be achieved by using SSL:

1. Identity Verification: A certificate signed to the server guarantees that the information the browser received is from the expected domain, more specifically, our online payment system. And, it is also guaranteed that the data sent from user to the server is useless if it is sent to a malicious third party since it is encrypted.
2. Data Integrity: The data sent between the systems is highly encrypted, and it is almost impossible for a malicious third-party to hack a connection and therefore prevent the man-in-the-middle attack.

(2) Challenge-Response Design



Challenge Response design is a very useful authentication design protocol in which one party presents a challenge and the other party must provide a valid answer to be

authenticated.

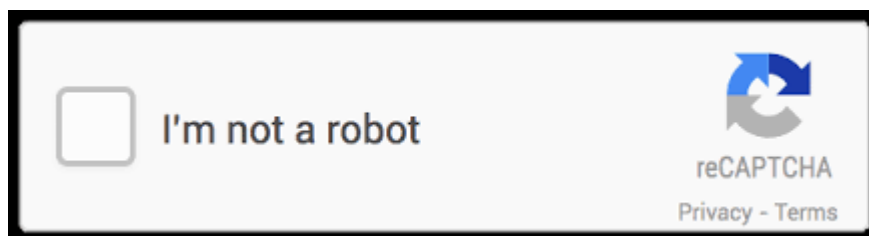
In our approach, we use an advanced Challenge Response Design with salt introduced. In the standard approach, the system needs to store the origin password, however, this approach will cause huge security issue when the database is disclosed, the origin password will be disclosed. Therefore, it is necessary to store the password in hashed and salted means. While this will also cause problems when implementing challenge response design since both client and server should both has the secret which is represented in password. Therefore, it is not easy to implement both hash-salt secret and challenge response design.

Hence, the improved challenge response design is used, and its basic workflow is divided into two parts, signup and login, when the user signup an account, the system will randomly generate a salt and hash the password with the salt, and therefore get a key value stored in the database. When the user login, the step is described as follow:

1. The client posts a request for acquiring the challenge.
2. The server randomly generates a secret key and encrypt it with the key stored in the database which is the hashed salted password and finally get the challenge value.
3. The server responds with the request with the challenge and salt used for the specified user.
4. The client uses the salt to hash its password using the same hash function with the server, and decrypt the challenge value using the key, where the client will get the secret key generated by the server
5. The client posts the login request with id and secret key to server, if the secret key matched, the server will store the user in the session and return with a status 200 OK.
6. The client reserved the response and take actions.

In this advanced challenge response design, we take the hashed salted password in to consideration and improve the security of the system.

(3) Google reCaptcha



Google reCaptcha provides an excellent approach against spam and abuse. The detailed version used is reCaptcha v2. It uses advanced risk analysis techniques to tell humans and bots apart, user only needs to click on the reCaptcha checkbox to distinct from bots. The detailed workflow is as follow:

1. We first acquire a pair of public key and private key from google reCaptcha website, with the public key on the client side and the private key stored in server side.

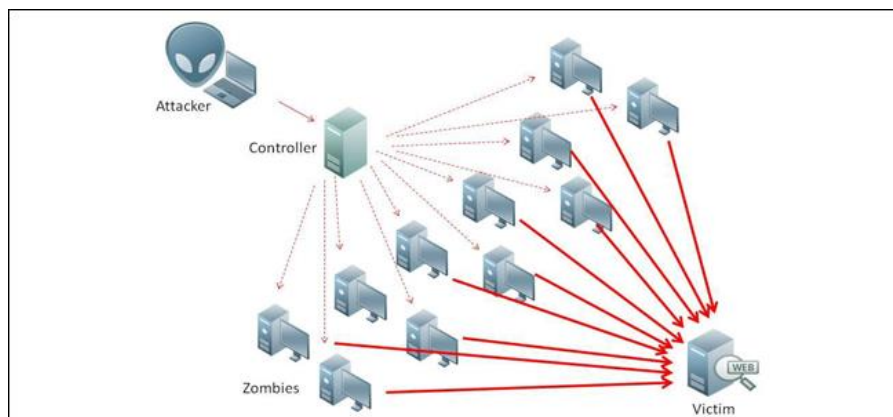
2. In the login page, there will be a reCaptcha checkbox, and the value of the reCaptcha will be saved in a hidden field.
3. When user click login, after the challenge response step, the reCaptcha value will be sent to the server side.
4. On the server, we send a request to Google reCaptcha API to identify whether the user has correctly pass the reCaptcha test.
5. If the API returns a correct result, the session will be recorded, and the user logged in successfully.

(4) Front-end/Back-end Verification

For data send to the server, it is needed to guarantee that all the data is in usable format which will not cause crash or problems. In our design, all the data in the fields is both checked by front end and back end.

For front end, there will be jQuery to parse and verify the input and for back end, there will also be a check procedure to prevent malicious attackers to post malicious or false data to the server.

(5) Prevent from DDoS Attack



For DDoS Attack, the google reCaptcha is the first wall which prevent user from infinite login. While there are other methods to prevent from DDoS Attack, in our approach, we also use the 'DDoS' nodejs package to auto detect DDoS Attack and prevent the server from DDoS attack.

(6) Protect Transaction information

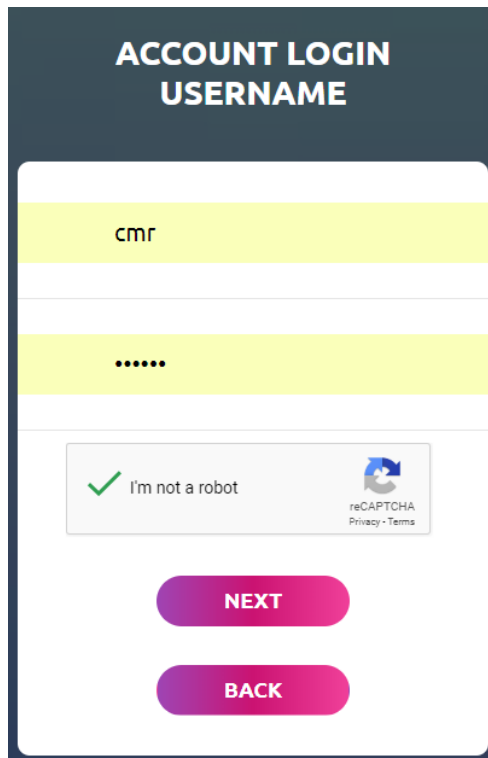
For each transaction made by the merchant and customer through the server, it is well protected. The detailed workflow is described as follow:

- i. First of all, the merchant needs to register an account in our website, and for each item he sells, he needs to assign a unique itemid for the item which is used to further verification.
- ii. The format of each pay link of the goods will be 'https://pay.chemry.me/pay?id=itemid&user=merchant_username&amount=good_price'.
- iii. After the customer click on the pay link, a pid will be generated to store the information which protects the transaction will not be injected through the login page, the login page link will be

‘https://pay.chemry.me/login?pid=transaction_pid’.

- iv. If the user did not login, he is required to login and continue his payment, however, if the session in our server site records the login status of the user, he will be directly leading to the pay page.
- v. The record will be stored as an activity if the user did not pay immediately, which ensures the whole procedure is success.

(7) Access control

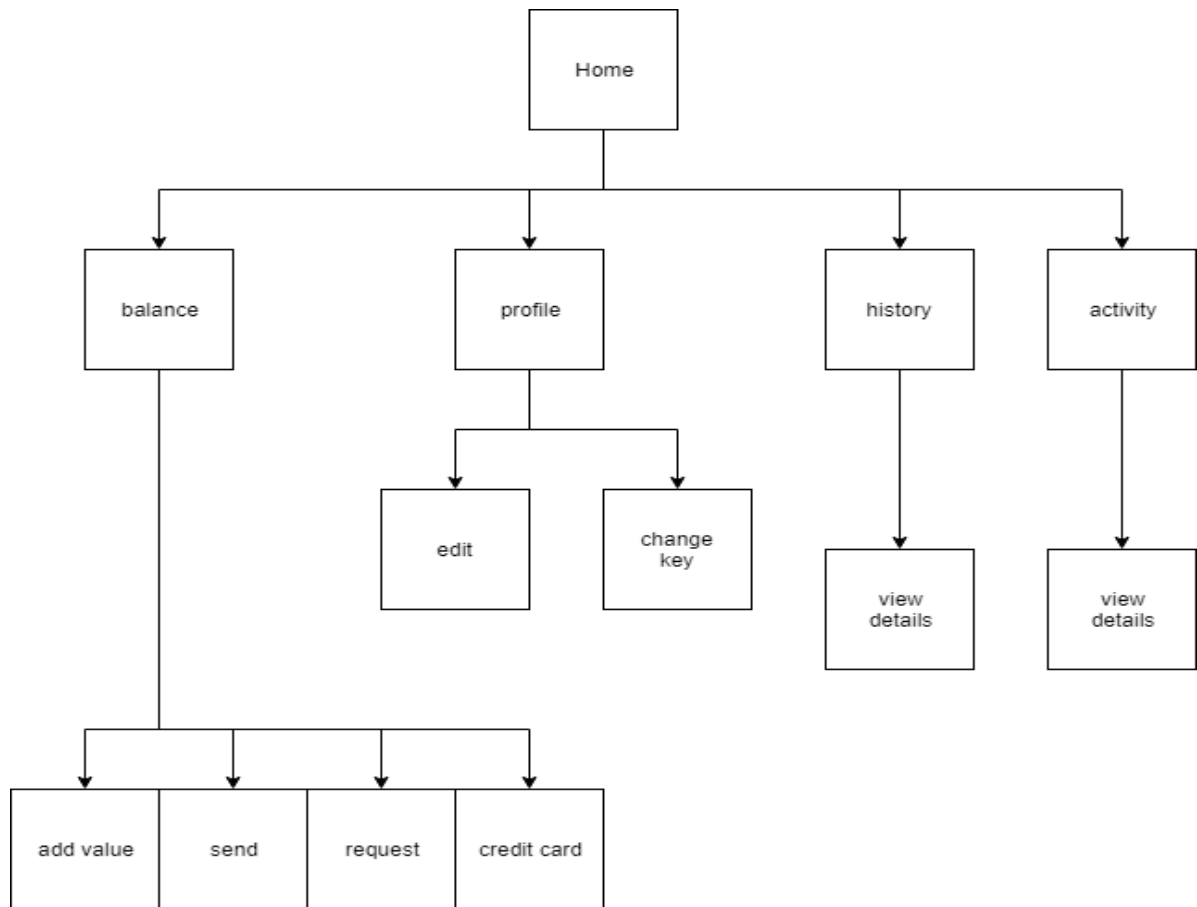
A screenshot of a web form titled "ACCOUNT LOGIN" with a subtitle "USERNAME". The form has a dark blue header. Below the header, there are two input fields: the first contains the text "cmr" and the second contains six dots. Below these fields is a reCAPTCHA widget showing a green checkmark and the text "I'm not a robot". To the right of the reCAPTCHA is a small icon of a person and the text "reCAPTCHA Privacy - Terms". At the bottom of the form are two pink buttons: "NEXT" and "BACK".

Access control is the selective restriction of access to the website page resources, while in our system, non-login user will be restricted to most of the page resources. In detail, only user can visit his own home page and only the restricted information will be provided.

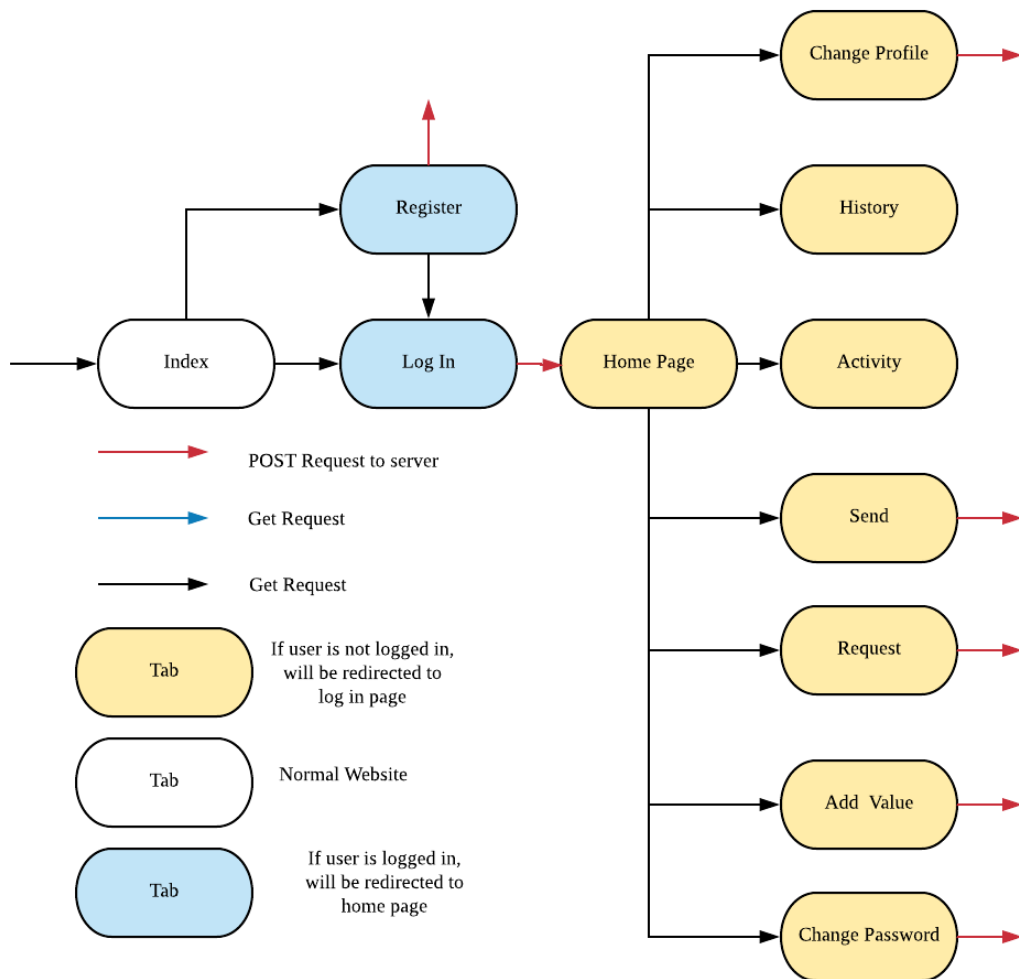
If the user try to access the page that he was not granted the permission to access, he will be redirect to the home page and no actual page will be sent. By using the methods of access control, the security of the website will be improved in huge manner.

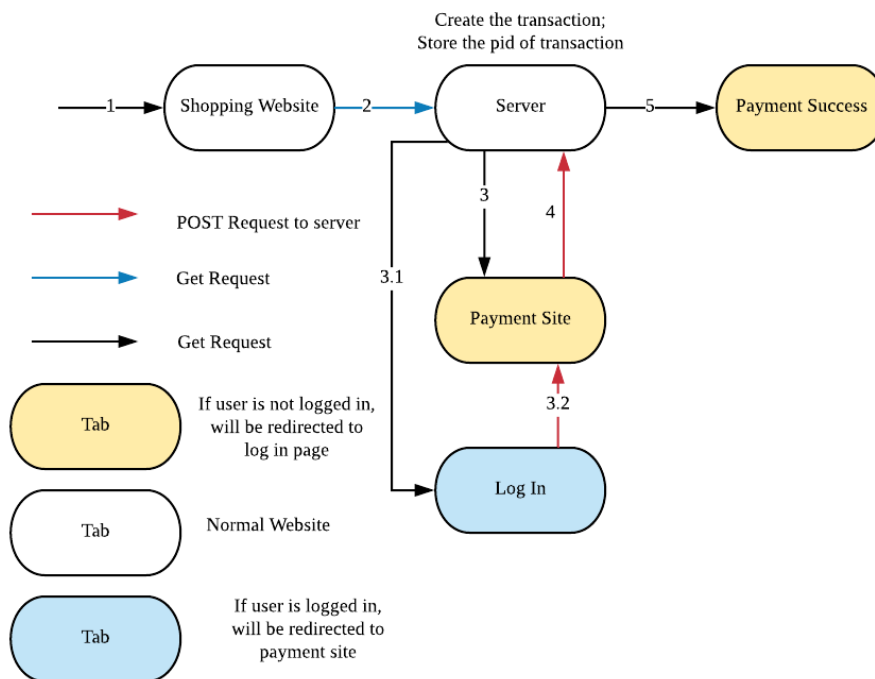
Website Structure and Security

(1) Website Structure



(2) Security Structure





Link 1:

Directed to the shopping website;

Link 2:

A payment request with parameters to be sent to the server.

(e.g. <https://pay.chemry.me/pay?id=123456&user=cmr&amount=23>)

The server first generates a payment id *pid* corresponding to this payment, then judge whether the user has logged in to the website or not.

Link 3.1, Link 3.2:

If the user has not logged in, he would be directed to the log in website; after he has logged in, he will be directed to the payment site with the *pid*.

Link 3:

The user is directed to the payment site with *pid*.

Link 4:

The server check with the *pid*. If the *pid* is existed and the user has entered the correct password, the payment would be success. If the *pid* does not exist or the user password is incorrect, the payment process could not be completed.

Application Installation Guide

First of all, To use SSL/HTTPS, you need to acquire the privatekey, certificate and ca_bundle, put them to the specified path.

Copy all the pass to the server, run:

1. npm install
2. node app.js

The server will run automatically.

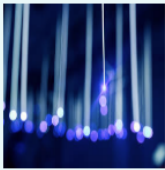
Use Cases

I. The usage case of payment methods.

a. The usage case of payment methods are divided into two situations. The first way of paying is in the situation that we trend to pay for a commodity from third-party website. This method is more complex and need our system to determine whether the user has log in or not. Details is as the following part:

User firstly uses third party shopping and finding out a commodity which he trends to buy. On that website, there should be a brief introduction of the goods with the price of it, and following should be a button which is a link that would lead you to our page.

Purchase



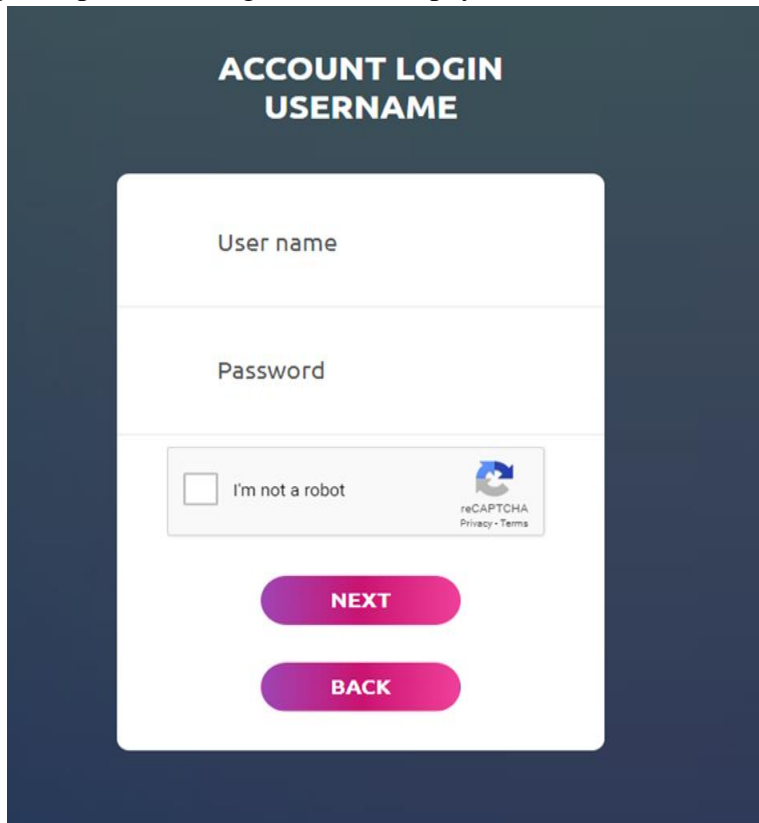
cmr
\$23

Purchase

Cancel

When the user decides to pay for the goods, he needs to press the button and will come to our identity page. There will be three cases relatively. First case is that the user has no accounts of our web, then he can choose to sign up for a new account for that payment.

After he signed up, he should go back to the payment URL and then login the account he

A login form titled "ACCOUNT LOGIN USERNAME" in white text on a dark blue background. The form is a white rounded rectangle containing three input fields: "User name", "Password", and a reCAPTCHA checkbox labeled "I'm not a robot". Below the inputs are two pink buttons labeled "NEXT" and "BACK".

**ACCOUNT LOGIN
USERNAME**

User name

Password

☐ I'm not a robot

reCAPTCHA
Privacy • Terms

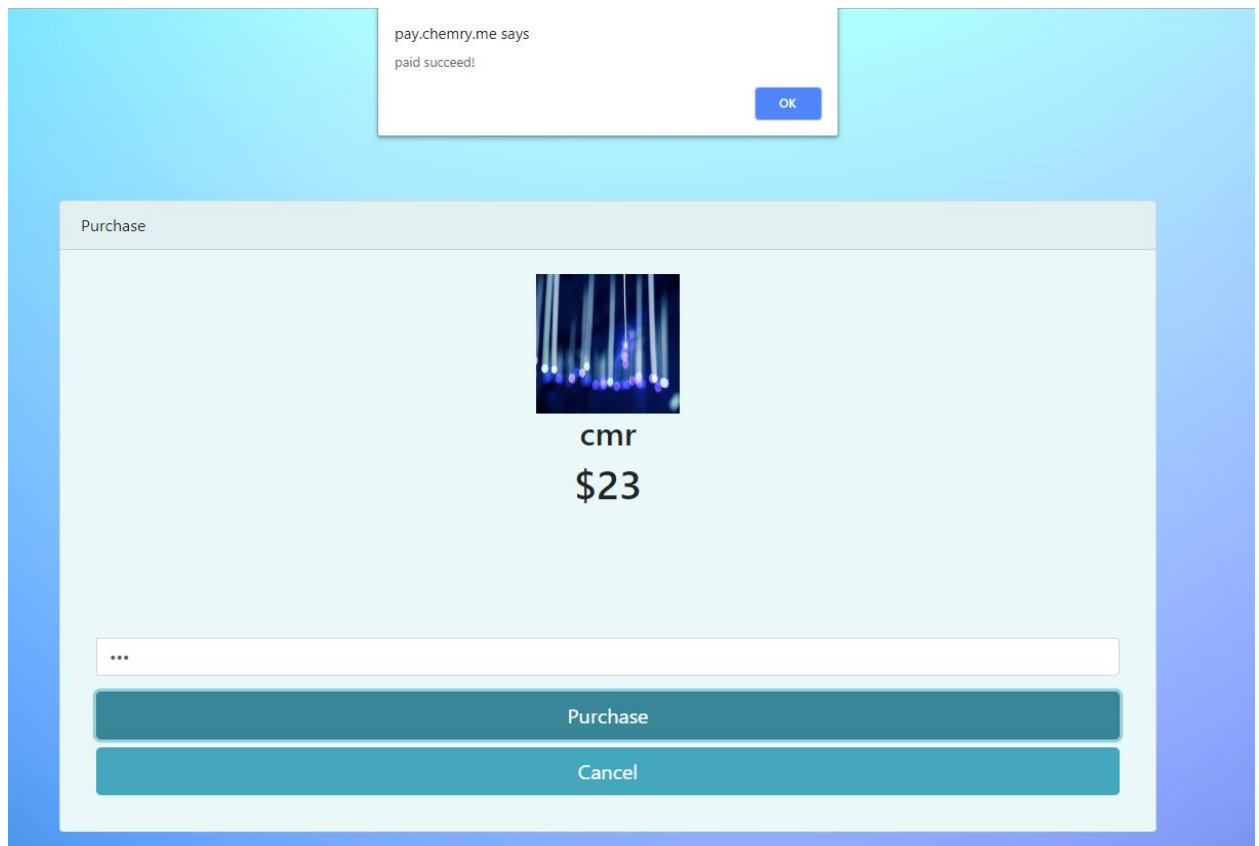
NEXT

BACK

registered.

Then he will successfully login and the pay website from our homepage will appear. Next, he could double check the detail of the payment and press the pay button. After pressing the Button, the payment will go end and the record of the trade will be displayed on the history

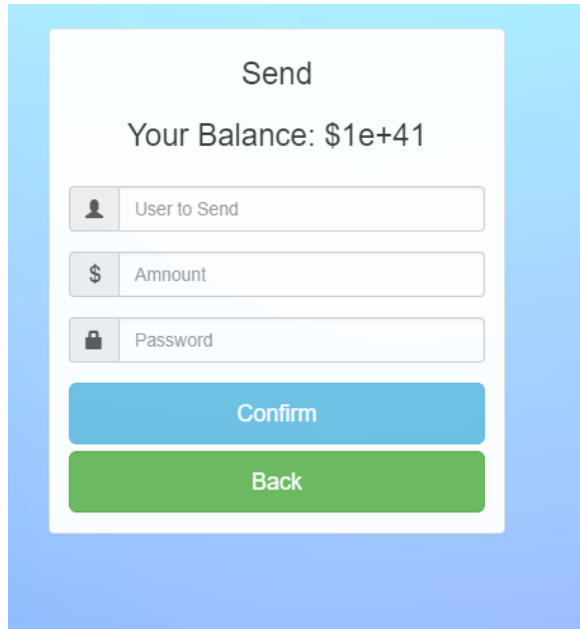
parts of his homepage.



Second is that the user has not login, same as the former case, he should choose login button instead of signup button, after he successfully login, the process of the next steps will be the same as first case. Third case is the simplest case that he will skip all the steps including login or signup, directly go to the payment page in our website and the following steps are same as other cases.

b. The second way of paying is in the situation that we trend to pay for a commodity from our own pages. The button of paying function is called send. It is used to send money from one user to the other user, it is internal trade may not relate to third part website or clients. The steps are as below:

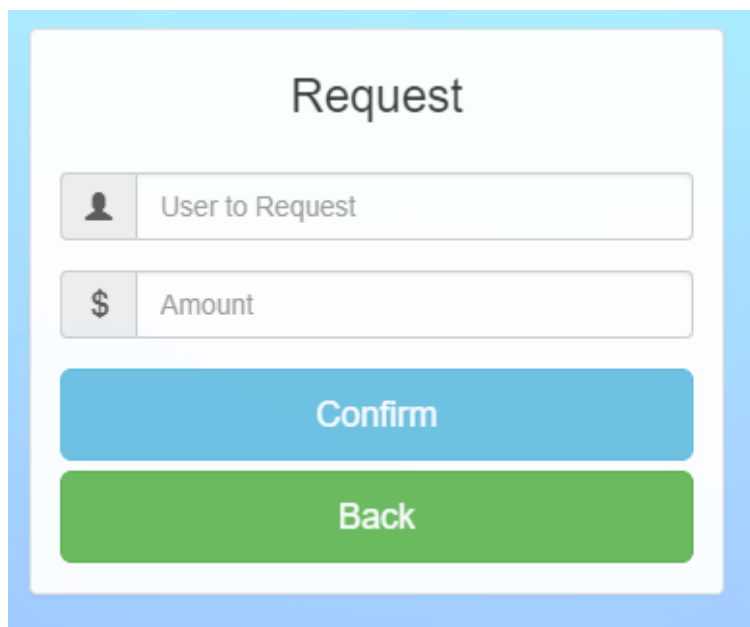
On the homepage, the user is asked to send money to the other one, he should click the button “send” in the balance part. When he clicks the button, he would go to the sending target information page and input the target username and the total amount of the funds. After that he input his password and press confirm button. Then a Pop-ups will appear and show the results: If the number of funds is less than your total balance, it will return success, or it return bad and the payment is invalid.



The 'Send' interface is a white rectangular box with a light blue border. At the top, it says 'Send' in bold. Below that, it displays 'Your Balance: \$1e+41'. There are three input fields: 'User to Send' with a person icon, 'Amnount' (misspelled) with a dollar sign icon, and 'Password' with a lock icon. At the bottom, there are two buttons: a blue 'Confirm' button and a green 'Back' button.

II. The usage case of receive money.

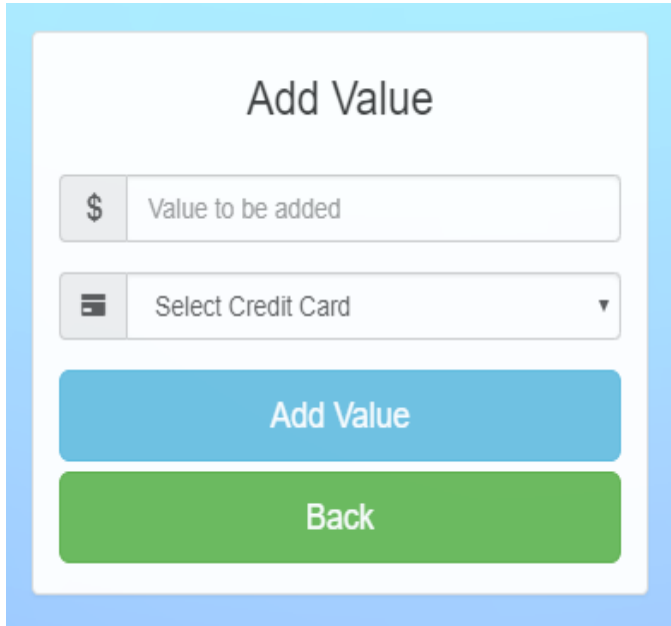
Receiving money must need users to login. When the user logs in and redirects to the homepage, if someone want to buy things from him but he did not find the link of the commodity, then you can send a request to him so that he can pay for it through the request. First, the user should click request button and go to the request page, there will be two input fields which are target and money number respective. Fill the information and press confirm to finish the request send. The object will correspondingly receive a link in his activity page waiting to be confirmed.



The 'Request' interface is a white rectangular box with a light blue border. At the top, it says 'Request' in bold. Below that, there are two input fields: 'User to Request' with a person icon and 'Amount' with a dollar sign icon. At the bottom, there are two buttons: a blue 'Confirm' button and a green 'Back' button.

III. The usage of value add.

Press value button, choose a credit card, filling the number user wants to add to the account.



Add Value

\$ Value to be added

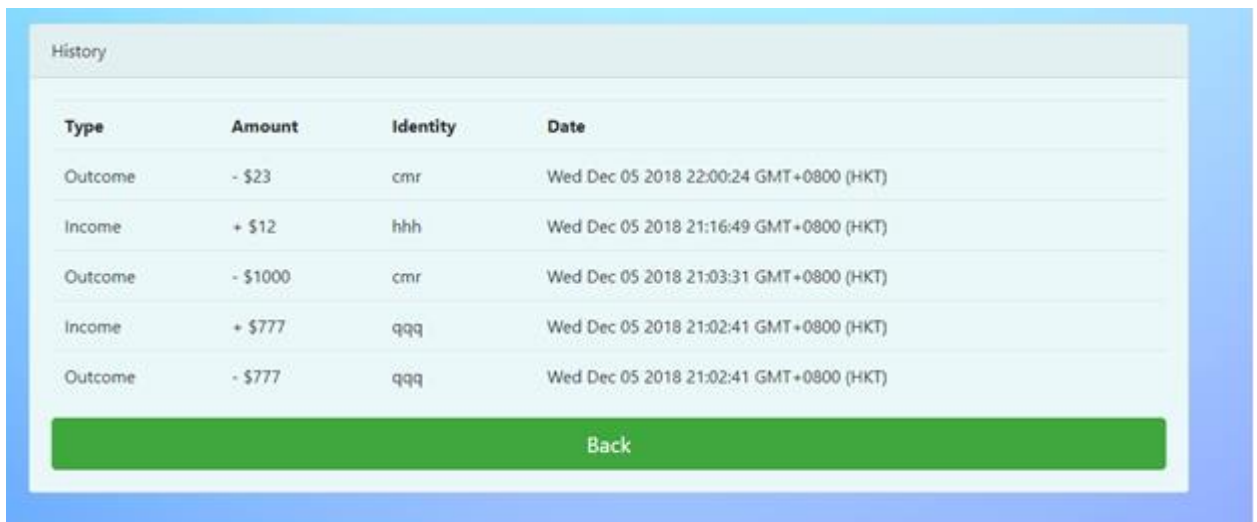
Select Credit Card ▼

Add Value

Back

IV. The usage of viewing history

Review the history of your past activities.



Type	Amount	Identity	Date
Outcome	- \$23	cmr	Wed Dec 05 2018 22:00:24 GMT+0800 (HKT)
Income	+ \$12	hhh	Wed Dec 05 2018 21:16:49 GMT+0800 (HKT)
Outcome	- \$1000	cmr	Wed Dec 05 2018 21:03:31 GMT+0800 (HKT)
Income	+ \$777	qqq	Wed Dec 05 2018 21:02:41 GMT+0800 (HKT)
Outcome	- \$777	qqq	Wed Dec 05 2018 21:02:41 GMT+0800 (HKT)


Back

V. The usage of receiving request

Related part II action, the part of activity is used to receive the request information. When view the activity, press the text link then there will be a billing detail page of the sender. If user admit it and he should input his password and confirm. The request and activity interaction will totally terminate.

Activity				
Type	Amount	Status	Name	Date
Received Request	\$23	Finished	cmr	Wed Dec 05 2018 21:59:38 GMT+0800 (HKT)
Received Request	\$23	Pending	cmr	Wed Dec 05 2018 21:58:23 GMT+0800 (HKT)
Received Request	\$23	Pending	cmr	Wed Dec 05 2018 21:57:56 GMT+0800 (HKT)
Received Request	\$null	Pending		Wed Dec 05 2018 21:57:30 GMT+0800 (HKT)
Received Request	\$null	Pending		Wed Dec 05 2018 21:57:23 GMT+0800 (HKT)
Received Request	\$12	Pending	hhh	Wed Dec 05 2018 21:15:29 GMT+0800 (HKT)
Back				

Purchase



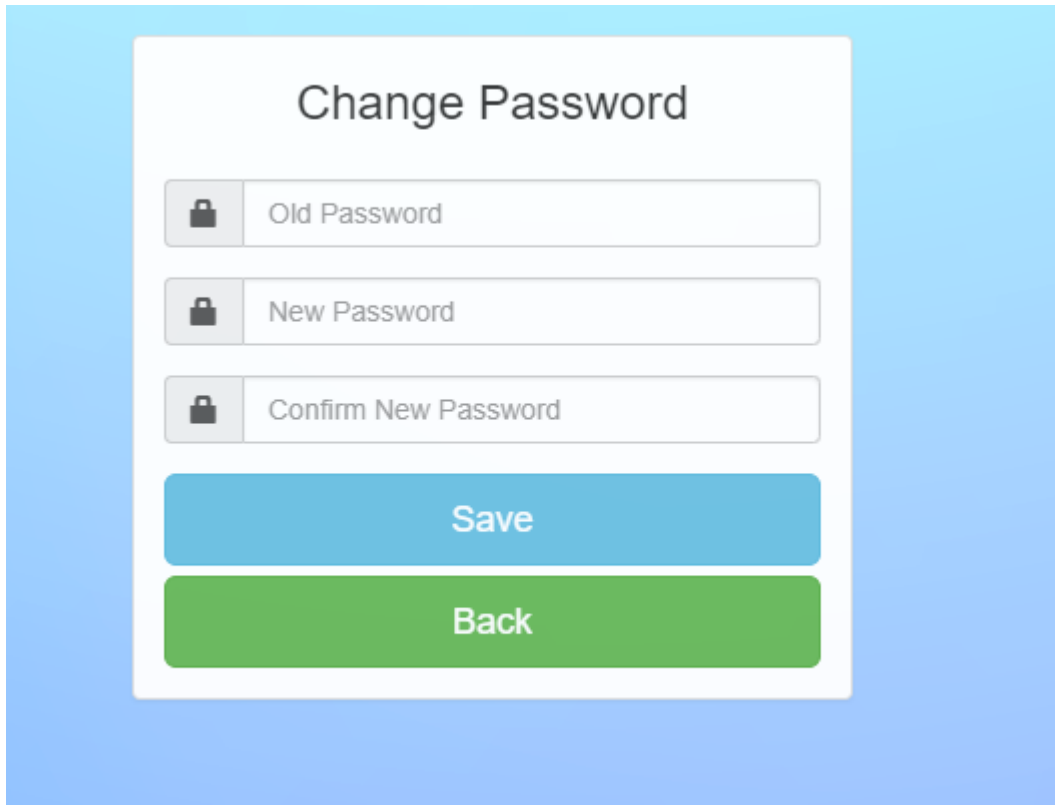
hhh
\$12

Purchase


Cancel


VI. The usage of change password


The button change password give user permission to change password

A screenshot of a 'Change Password' form. The form is white with rounded corners and is centered on a light blue background. It contains three input fields, each with a lock icon on the left and a label: 'Old Password', 'New Password', and 'Confirm New Password'. Below the input fields are two buttons: a blue 'Save' button and a green 'Back' button.

Change Password

 Old Password

 New Password

 Confirm New Password


Save

Back


VII. The usage of edit profile

The button of edit profile is used to change your personal information.


Update my information




qqq




qqq



755749011@qq.com



qqq



qqq

Save


Back

VIII. The usage of manage credit cards

User can add or delete credit cards through page after pressing the credit cards button.

Credit CardInformation

+ Add New

Card Number	Name	Expiration Date	Verification Number	Actions
2222	2222	2018-12-01	11	

Bibliography

Hsieh, C. T. (2001). E-commerce payment systems: critical issues and management strategies. *Human Systems Management*, 20(2), 131-138.

Linck, K., Pousttchi, K., & Wiedemann, D. G. (2006). Security issues in mobile payment from the customer viewpoint.

Kim, C., Tao, W., Shin, N., & Kim, K. S. (2010). An empirical study of customers' perceptions of security and trust in e-payment systems. *Electronic commerce research and applications*, 9(1), 84-95.

Kalakota, R., & Whinston, A. B. (1997). *Electronic commerce: a manager's guide*. Addison-Wesley Professional.

McDowell, M. (2004). Understanding denial-of-service attacks. *National Cyber Alert System, Cyber Security Tip ST04-015.2004*.

Xiaohu, X. U. (2009). *U.S. Patent Application No. 12/436,929*.

// <https://www.websecurity.symantec.com/security-topics/what-is-ssl-tls-https>

// https://en.wikipedia.org/wiki/Challenge%20%80%93response_authentication

Swift, M. M., & Shah, B. (2002). *U.S. Patent No. 6,377,691*. Washington, DC: U.S. Patent and Trademark Office.

// https://en.wikipedia.org/wiki/Salted_Challenge_Response_Authentication_Mechanism

Newman, C., Menon-Sen, A., Melnikov, A., & Williams, N. (2010). *Salted challenge response authentication mechanism (SCRAM) SASL and GSS-API mechanisms* (No. RFC 5802).