

Second-Hand Shopping Website report

Node.js Project

HAOYU WANG

Software Development
Bsc(Honour)



Computer Science
GMIT
IRELAND
14 /APRIL/2016

Contents

1 Abstract	4
2 Introduction	5
2.1 Idea Raised	5
2.2 Website Prospect forecast	5
2.2.1 Biggest part of consuming	6
2.3 The important thing of Online Shopping is Security	6
2.3.1 Lack Of Full Cost Disclosure	7
2.3.2 Quality is not guaranteed	7
2.3.3 Personal information leakage	7
2.4 Design ideas	7
3 Methodology	8
3.1 How i did my project	8
3.1.1 Developing Method	8
3.1.2 Discuss requirements of Website	8
3.2 Development tools	8
3.2.1 The hardware environment	8
3.2.2 Software Environment	9
3.2.3 Komodo Edit [12]	9
3.3 Platform Brief	9
3.3.1 GitHub profile [11]	9
4 Technology Review	11
4.1 Node.js	11
4.2 Database	11
4.3 JSON	12
4.4 Express.js	12
4.5 Mongoose	12
4.6 Ajax	14
4.7 Crypto-js	14
4.7.1 Support Algorithms	15
4.8 Session [4]	15
4.9 Nodemailer	15
4.10 Autocomplete.js	15
4.11 PayPal sandbox	16
4.12 Base64	16
4.13 date-utils	17
4.14 Regular Expression[18]	17

5 System Design	18
5.1 Language Choose	18
5.1.1 JavaScript's features	18
5.1.2 Object-Oriented	18
5.1.3 Variable and Object	19
5.1.4 Cross-Platform	19
5.2 Node.js	19
5.2.1 Node.js's advantages on web development	19
5.2.2 Powerful Institutions are used Node.js	20
5.3 Architecture Review(uml)	20
5.3.1 system Architecture(uml)	20
5.4 System Functions Review(uml)	21
5.5 Security System	24
5.5.1 Consumer's Password	24
5.5.2 Session and Cookies	25
5.5.3 Database Security Design	25
5.5.4 Face to Face Trade	26
5.6 Database design	26
5.6.1 Database Users	26
5.6.2 DB:webSite	26
5.6.3 DB:webSite-app	28
5.6.4 webSite : Orders	28
5.6.5 Code-create tables	28
5.7 Core Function	29
5.7.1 Hottest Item	29
5.7.2 Search Function	30
5.7.3 Session	30
5.8 Appearance Design	30
5.8.1 Mainpage	30
5.8.2 User Management Page	32
5.8.3 Items Page	33
6 System Evaluation	38
6.1 Advantages	38
6.2 Limitation	38
6.3 Future Development and Optimization	39
7 Conclusion	40
A Appendix	42

Chapter 1

Abstract

In this, project I have used Node.js, Express.js framework, Mongoose for MongoDB and the middleware such as crypto, autocomplete and connect.

Nowadays people have certain degree of waste in shopping. Sometimes they are impulsive to buy items when the shop has discounts or privileges.

And there also have a different situation like me an international student who need study and live in a foreign country for a few years, and in the end of studying when they need back to home country a lot of living products must cannot take to home. So, I built a shopping website to help people who need to sale second-hand item or buy those use low-price. And this website I used Node.js and MongoDB to realize all functions I need. Node.js is a new software development platform, it is not only a separate language, but also with Python, Ruby, as not only a language but also internet. Node.js the JavaScript from your browser ported to the common server. So that JavaScript can be achieved such as file systems, operating systems, network communications and other functions. Node running on Chrome V8 engine, asynchronous I / O architecture and design and single-threaded event-driven model, and built-in HTTP server it is mainly used to build high-performance, high-concurrency, high scalability of the server [14]. Develop a Web application, the most important is the database using, MySQL, PHP was best partner in the past, and now there is MongoDB and Node.JS make the best combination. MongoDB have some special features that is a new NOSQL Data Base and Simple use, free, open source and ready to download [13].

Based on those points, this website I use a flexible and small frame named Express.js that can make Node.js easy and improve application running efficiency in the same time. Express.js is a frame in Node.js that be developed to solve website problem and based on the existing framework for analyzing different platforms to extract feature points of the frame. These include routing and forwarding function point mapping, property injection, Cookie achieve , Session realization.

Key Point: Shopping, Second-Hand Product, Node.js, MongoDB, Express.js.

Chapter 2

Introduction

2.1 Idea Raised

This is a Node.js Project that for Final year project named Second-Hand shopping website, this idea is emerging from myself experience that when I first time come to Ireland I can't adapter to different eating habits and culture. I'm extremely want to cook Chinese food, but many Chinese cookers (some Chinese student took it from home when they come to foreign country) I can't buy in here. So, I raised an idea that if I can buy those second-hand cooker from shopping website must good, and when I leave this country I can sale most living product to other people who need it.

On the other hand, thinking this from Market side. Impulse buying (Research of Impulse Buying [7]) is a very common it happened around our life. I believe every have some similar experience when you bought some products and arrive home fell this is not really useful in home. But you don't want throw it then store it. This kind of behaviour caused great waste, so my website also can give them a place to dispose some useless product and get a suitable return.

Thirdly, it very hard when young people just start them first job and move out family live with themselves, many living product need to buy such as sofa, ice-fridge, table, or TV. Some people don't have enough money to buy new items, my website will become a place help them use litter money to get those furniture. WWW From birth to present, Impact on people's lives at an alarming rate , Revolution in e-commerce on the Internet came into being. Online shopping system is through Web Online purchase system, This paper studies based on Node.js Design technology of online shopping system.

2.2 Website Prospect forecast

According Data and trend, Stacey Rudolph [9] was written an article named The Future of Online Shopping is Bright—Statistics and trends. The main content in article is Retail worldwide onlone retail spending is expected to grow rapidly over the next few years. There have a picture below see figure 2.1:



Figure 2.1: Worldwide retail online sales

2.2.1 Biggest part of consuming

Paco Underhill [2] the author of a book named Why we Buy said “Two-thirds of the shopping that goes in in malls is impulse buying” and “Actually two-thirds of the economy is impulse buying” Many people fall victim to the last-minute impulse purchase. In fact, 90 percent of shoppers buy items not on their shopping list. Therefore, we can see online shopping already become a modern people’s lifestyle, and I’m firmly believe Second-hand Shopping website will have a good future, because It not only match most people’s shopping habit but also can help them to dispose some useless furniture and get corresponding returns.

2.3 The important thing of Online Shopping is Security

In before, people start choose use online shopping the most important thing for them is time, there have a successful business man said: ”when I was younger was traded time for money, now I trade money for time. The reason that I like to shop online, because it saves time.” Many people said the most attractive thing let them shopping online is not the price in fact, but save time, delivery is the main reason let they choose to shopping online.

However, nowadays customers need to pay more attention on security when they are shopping because we are living in a technically advanced world and it leaves us vulnerable any time we are go online. Abe Garvet have a article to describe some risks when you shopping online.[15]

Fake On-line Reviews

A collection of five-star ratings can do amazing things for on-line sellers. As a result, some authors have confessed to posting fake reviews; So customers will easily be taken in those fake information.

2.3.1 Lack Of Full Cost Disclosure

Additional fee such as delivery fee, maybe hidden until you complete check out part. For example: when you onlin shopping and compare price between different shops, finally you find a prefect one have low-price and free delivery. But when you check out there have a line of small words said you need to make membership (\$ 45)to get free delivery, you did this. This meant the actual cost was 20% greater than I was lead to believe. Don't expect software to come with the computer.

2.3.2 Quality is not guaranteed

As is known to all, the quality is the consumer to purchase goods and services, the focus of the core factors and at present a lot of online consumers actually received goods are not in the eyes of goods, product quality is also often appear problem, it makes the security and reliability of network people on online shopping

2.3.3 Personal information leakage

There have some companies do not have completely security to protect consumers account detail, it let many hackers can get them information easily.

For example: if someone steals your debit card number, your entire checking account is exposed. Within minutes, identity thieves can drain your account and set you back financially for years

Finally it will let consumer get lost in economy.

2.4 Design ideas

Online shopping system makes full use of Brower/Server structure characteristics of implementation system ported to Internet functions, has the advantage of great convenience to the users of the system, reduces the regional restrictions and may not be shopping phenomenon.

And point the most important part of online shopping website design is security. So in second-hand shopping website I will pay more attention on security system design.

Chapter 3

Methodology

3.1 How i did my project

3.1.1 Developing Method

I used Node.js, Express.js and MongoDB to develop shopping website. Firstly I make a requirements analysis and then I use Rapid Prototype Model [17] create a template only have front-end.

3.1.2 Discuss requirements of Website

A good online shopping system, not only include an optimized back-end but also need a beautiful front-end, this system is consist by four part: Firs: the customers browsing module;

Second: application service module;

Third: the back-end database module;

Fourth: is a most important one, and back-end systems client management module. At the front desk of the modules contained in the registration, login, logout processing functions, and function of these three modules are included in the query. Modules in the background, only the administrator has the authority to login, the administrator can add product information, customer, you can edit details of the commodity and so on a series of functions.

I start make front-end (website framework) and thinking the necessary functions will used in my project. I create login, logout, account. And display some items on this page. Then start do the back-end.

3.2 Development tools

3.2.1 The hardware environment

Laptop configuration: CPU: I5-5200U 2.2GHz;

RAM: 8 GB; Memory: 512GB

Operation System: Microsoft Windows 10

3.2.2 Software Environment

Developing tool : KOMODO IDE 9

Browser: Firefox, chrome

Development Environment: Node.js

Database: MongoDB

3.2.3 Komodo Edit [12]

Download address: <https://github.com/Komodo/KomodoEdit>

Komodo Edit is a free text editor for dynamic programming languages. It was introduced in January 2007 to complement Active State's commercial Komodo IDE. As of version 4.3, Komodo Edit is built atop the Open Komodo project. Many of Komodo's features are derived from an embedded Python interpreter.

3.3 Platform Brief

3.3.1 GitHub profile [11]

GitHub is a shared hosting services that use the Git to share system projects. It were made by the GitHub Company's (called Logical Awesome) developers Chris Wanstrath, and PJ Hyatt, and Tom Preston-Werner write by Ruby on Rails. GitHub also offers paid accounts and free accounts for open source projects. According to the Git user survey, GitHub is the most popular of the Git access site.

GitHub Personal information library, and the Wiki page through a git repository you can edit these pages. As a distributed version control system, Git does not have a main library in concept, each copy out of warehouse can be used independently, and any difference between these warehouses can be merged together.

GitHub can host a variety of git, and provides a unified Web interface. But provide different service with others Code manage website such as Source Forge and Google Code, GitHub is unique to branch a project management is very simple, has a high degree of interoperability. If you want to contribute code for a project is an easy task: first on the project site of the "fork" button, then check out the code and to add modifications to separate code libraries, and finally through the built-in "pull

request” mechanism to apply to the project director code.

Chapter 4

Technology Review

4.1 Node.js

Node.js is a JavaScript running environment (runtime). In fact it is encapsulate Google V8 engine. The speed of V8 engine execute JavaScript is very fast, performance is very good. Node.js optimize some special cases, provides alternative API, makes the V8 running better in the browser environment. It also is a platform which based on Chrome JavaScript, be used to build network Application with fast response speed and expand easily.

Working principle see graph 4.1:

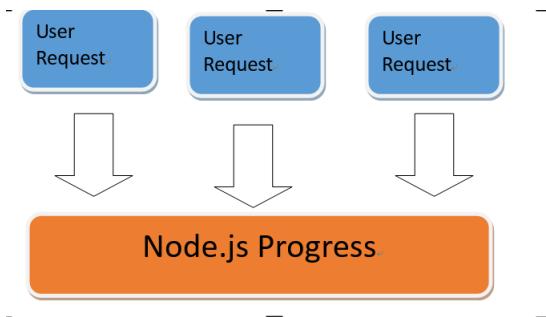


Figure 4.1: Shopping Cary Mini State window

4.2 Database

MongoDB was used to store all the data in shopping system include users' information and items' detail such as price, name, id and color.

MongoDB is classified as a NoSQL database and also is a cross-platform document-oriented database. [13] There have some features let me choose MongoDB: Support JSON-file (MongoDB calls the format BSON), it let me import data easy and fast than SQL or MySQL. It can import JSON file or CSV use a line of code, see code put below.

Setting code : /bin mongoimport -d cms -c -type csv -file c.csv -h localhost -port 11111 -upsert -f name

MongoDB have 6 main data type:
Number, String, Boolean, Array, Object, Null.

4.3 JSON

[6] JSON (JavaScript Object Notation) is a lightweight data-interchange format. Sometimes JSON is an open-standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is the most common data format used for asynchronous browser/server communication (AJAJ), largely replacing XML which is used by AJAX.

Exapmle of JSON file :

```
var jason =  
  "age" : "24",  
  "hometown" : "Missoula, MT",  
  "gender" : "male"  
;
```

And it also can include the Emoji character U+1F602 (cryface) FACE WITH TEARS OF JOY in JSON: { "face": "(cryface)" } or { "face": "\uD83D \uDE02" }

4.4 Express.js

Express.js is a simple and flexible web application framework in Node.js, it's not upgrade version of node.js but add a lot of powerful feature on web development.
Current Express.js Version: 2.14.7

Install code: npm install express
var express = require('express');

4.5 Mongoose

[8] Firstly, I have used MongoDB self to operate Database, it generate many problems such as

Error: db object already connecting, open cannot be called multiple times.
I waste a lot of time to solve this problem, finally I find the problem is sometimes I refresh webpage too fast and the Database didn't close will result in this problem.
There also have some others problem let me give up use mongodb self to control Db in Node.js.

I give a compare example below:

First step: npm install mongoose –save
var mongoose = require('mongoose');

```
mongoose.connect('mongodb://localhost/test');
Mongodb: every time operate DB need mongodb. Open (), and mongodb. Close () in the end.
```

```
User.get = function get(username, callback) {
  mongodb.open(function(err, db)
    if (err) {
      return callback(err);
    }
    Read users collection
    db.collection('users', function(err, collection) {
      if (err) {
        mongodb.close();
        return callback(err);
      }
      //find a file name is username in collection
      collection.findOne({name: username}, function(err, doc) {
        mongodb.close();
        if (doc) callback (err, doc);
        else callback (err, null);
      });
    });
  });
};
```

Mongoose operate DB: Only need a few lines code

```
User.get = function get(username, callback) {
  users.findOne({name:username}, function(err, doc){
    if (err) {
      return callback(err, null);
    }
    return callback(err, doc);
  });
};
```

And paste some mongoose method below:

Create :

```
DB.save(function (err) {
  if (err)
    console.log('meow');
});
Remove
```

```
DB.remove({name:"ccc"},function (err) {
  if (err) // ...
  console.log('meow');
```

});

Etc.

4.6 Ajax

[1] Ajax (Asynchronous JavaScript and XML) is a very powerful technology that exchange a small amount of data in the back-end let website don't need refresh of loading all data in the website again to update apart of website.

Ajax module compare with traditional module

Example of Ajax in Node.js, see graph 4.2

```
$ .ajax({
  url: '/itemupdate',
  type: 'POST',
  data: 'username=' + $("#username").val()
    +'&location=' + $("#location").val()
    +'&category=' +$("#category").val()
    +'&itemname=' +$("#itemname").val()
    +'&picture=' +encodeURIComponent(this.result).replace(/\+/g, '%2B')
    +'&phone=' +$("#phone").val()
    +'&description=' +$("#description").val()
  ,success: function(data) {
    if (data=='') {
      alert("success update");
      window.location="/item_category_add";
    }else{
      alert("update wrong");
      window.location="/item_category_add";
    }
  }
});
```

Figure 4.2: Ajax

4.7 Crypto-js

This is a modular in npm that offer methods to encrypt and decrypt information. Node using the OpenSSL library to realize its encryption technology, this is because the OpenSSL already is a widely used encryption algorithm. It includes similar MD5 or SHA - 1 algorithm, these algorithms you can use in your application.

And crypto modular use Asymmetric encryption that refers to both sides in a different KEY encryption and decryption expressly, communication both sides have their own public KEY and private KEY.

Easy to understand, for example, we are assuming that communication both sides respectively is A and B. A, have KEY_A1 KEY_A2, including KEY_A1 is a private key, KEY_A2 is a public key, B has the KEY_B1, KEY_B2, including KEY_B1 is B's private key, KEY_B2 is B's public key. The characteristics of public key and private key is, after any of them a plaintext encrypted, can only use the other one was able to solve. That is to say after plaintext encrypted KEY_A1, only can decrypt KEY_A2, and vice versa.

4.7.1 Support Algorithms

Crypto-js support many algorithms such as MD5, SHA-1, SHA-256, AES, Rabbit, MARC4, HMAC (HMAC-MD5, HMAC-SHA1, HMAC-SHA256), PBKDF2.

4.8 Session [4]

In a web application user's opinion, he opens a browser to access e-commerce site , log in and complete the shopping until you close your browser , this is a conversation . In web application developer's mind, they need to create a data structure to store the user's information when a user login, this structure also called session. So we talk about session time to pay attention to context. The article talking about is based on the HTTP protocol to enhance the ability of the mechanism web application or a scheme , it does not refer to a specific dynamic page technology , and this ability is to keep the state and to be called to keep session.

Session have many advantages in different part such as security, property, convince. And sometimes if we need session have long statement cycle, such as many web sites have a password the function of the automatic login within two weeks. Based on this requirement, the session must be looking for memory storage load, the database can provide the perfect solution. Here, I choose MongoDB as database, as a NoSQL database, data object database - collection - the basis of it the document object model is very intuitive and easy to understand, in view of the node.Js also provides rich drives and API. Express framework provides for directing a middleware: connect - mongo, we just at the time of mount session in the options to mongo parameters, program is running, express the app will automatically managing the session storage for us, update, and delete.

Install Session: `npm install express-session`

And install a middleware: `npm install connect-mongodb`

4.9 Nodemailer

This is a modular in node.js which can send email automatically. It's very helpful in my shopping website I can send email to consumer when he or she successed register a account. And also send email to them when they finish an order.

Install code: `npm install nodemailer --save`

4.10 Autocomplete.js

[10] This is a JQuery technic, it can fill the search box automatically realize the function like Google search.

The example picture 4.3 show below:

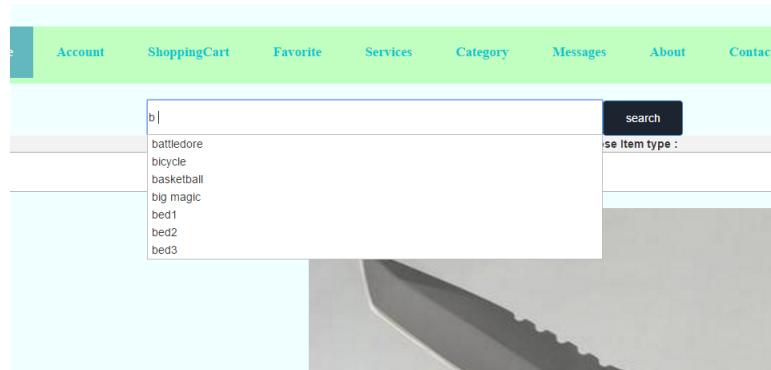


Figure 4.3: Autocomplete fill

License:

Ajax Autocomplete for jQuery is freely distributable under the terms of an MIT-style license.

Copyright notice and permission notice shall be included in all copies or substantial portions of the Software.

4.11 PayPal sandbox

PayPal Sandbox environment is an independent Environment and in a Sandbox environment does not involve actual money transactions. Its purpose to provide developers and integration environment, The Sandbox allows you to have the opportunity to before transplantation applications to a true PayPal to you the whole process of integration testing, avoid deployed to the possible problem in real environment.

4.12 Base64

Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation. The term Base64 originates from a specific MIME content transfer encoding.

The Node has a binary Buffer, the pseudo class provides a series processing of binary data API, simplifies the need to deal with the task of binary data. The length of the buffer is determined by the length of the byte data, and you can random set and get bytes of data in the buffer.

```
var utf8String = 'my string';
var buf = new Buffer(utf8String);
var base64String = buf.toString('base64')
```

4.13 date-utils

Date-utils is a package in npm. In Nodejs get current system time is hard then other programming languages and Node provide a class to display time.

```
install : npm install date-utils
require('date-utils');
var dt = new Date();
console.log(dt.toFormat("YYYY-MM-DD HH24:MI:SS"));
```

4.14 Regular Expression[18]

In theoretical computer science and formal language theory, a regular expression (sometimes called a rational expression) is a sequence of characters that define a search pattern, mainly for use in pattern matching with strings, or string matching, i.e. "find and replace"-like operations. The concept arose in the 1950s, when the American mathematician Stephen Kleene formalized the description of a regular language, and came into common use with the Unix text processing utilities ed, an editor, and grep, a filter. I use Regular Expression to match some word that on search function which can get more similar words. And it also helpful on MondoDB to search data.

Chapter 5

System Design

5.1 Language Choose

5.1.1 JavaScript's features

[3]

Node.js is based on JavaScript design

JavaScript is a scripting language that based on the object and event driven, and it also good at security capability, JavaScript is dedicated to develop INTERNET client and server applications, it can be embedded into HTML file easily, and through the built-in interpreter of the JavaScript in the browser to perform. Using JavaScript, the browser can respond to the needs of users events without back data through the network. In this way, the user's data can be directly by the client application processing. It also can easily realize the interaction with network customers, and make the web becomes vivid.

Forexample: when you visit a website, JavaScript can give responses immediately by move mouse or click some button or other component in this page or drag windows and etc.

5.1.2 Object-Oriented

Talking about the object-oriented programming language, most people immediately think of C++ and Java that a strongly typed language, Python, and Ruby, scripting languages, their common feature is a class-based object-oriented. And when it comes to JavaScript, it's hard to make people think its object-oriented programing language, because it does not have any classes. Although JavaScript does not have classes, but JavaScript is an object-oriented language. JavaScript object only, and an object is object, not an instance of the class. Because most are based on class of objects in object-oriented languages, it is easy to misunderstand the concepts of an instance of the class with the object. The definition of object is an instance of the class maybe correct in most other languages, but it's not for JavaScript.

5.1.3 Variable and Object

The object in JavaScript actually is a related Array that made by property which consisted by name and value, the value type can be any data type, or a function, and other objects. JavaScript has a special characteristic of functional programming, so function also is a variable, most of the time same as general data types. In JavaScript, use the dot operator is equal with an associative array reference, meaning that any object (including the this pointer) can use either mode. Are the benefits of using an associative array, if in a case but you do not know the names of the object, you can use an index as an associative array.

5.1.4 Cross-Platform

JavaScript scripting language is not dependent on the operating system, only need the support of the browser. So JavaScript can be used to any machine, after writing the premise on the machine on browsers support JavaScript scripting language, JavaScript is supported by most browsers.

5.2 Node.js

[14]

Use Node.js with MongoDB to develop this website because human being already come in Big Data, huge amount data need processing through browser. So the efficiency especially important in a web application, and now Node.js can provide a high-efficiency platform to develop web.

5.2.1 Node.js's advantages on web development

Node.js use event-driven, asynchronous programming, and network design services. In fact, anonymous functions and closures characteristics JavaScript is ideal for event-driven, asynchronous programming. And JavaScript are easy to learn, a lot of front-end designers can quickly get started doing back-end design.

Node.js non-blocking mode processing to bring in relatively low system resource consumption at high performance and outstanding load capacity, very suitable for mid-tier service used to rely on other IO resources. And Node.js lightweight and efficient, it can be considered the perfect solution for data-intensive real-time applications distributed deployment environment under. Node is very suitable for the following: In response to a client before, you may be expected to have a high traffic, but the required server-side logic and processing is not necessarily a lot.

5.2.2 Powerful Institutions are used Node.js

1. Pay Pal [5]

There have a paper was written by Jeff Harrell show the sharp increasing of efficiency, amount of code and performance after PayPal use Node.js to develop application.

There have a competition in PayPal engineers, one team with five engineers working on java application, another team with two people working on Node.js application and they are need to realize same functions. In the end, when they ran the test cases ,the node.js app was:

- Built almost twice as fast with fewer people
- Written in 33% fewer lines of code
- Constructed with 40% fewer files

2. Twitter queue

Imagine companies such as Twitter, it must receive tweets and written to the database. In fact, almost every second thousands tweet reached, the database cannot be processed in time to write the huge number of message in peak hours. Node.js has become an important part of the problem solution. As you can see, Node can handle tens of thousands of tweet. It can write them to a memory queuing mechanism (such as memcached) quickly and easily, a separate process will written those tweet to the database. Node's role here is to quickly collect tweet, and this information is passed to the process and the other for writing. Imagine another design (regular PHP server will attempt to process its own writes to the database itself): Each tweet will lead to a short delay when writing to the database because the database call blocking passage. Since the database delay, machine could only handle 2000 incoming tweet every second. 1 million per second tweet need 500 servers. Instead, Node can handle each connection without blocking the passage, it is possible to capture as many tweets. A tweet can handle 50,000 machines in just 20 of Node server.

5.3 Architecture Review(uml)

5.3.1 system Architecture(uml)

I used a minimal and flexible Framework to make my website is Express.js, and there also have some functions it need have and show those below:

(1) Route maps.

Different types of requests can be mapped to different business logic, which should include requests for static data and data requests for the distribution.

(2) The Cookie feature.

Meet the needs of certain data need to be stored in the client browser.

(3) Session function.

Realize Session conversation by Cookie, and to store a small amount of data through-

out the session.

(4) Property injection function.

Use JavaScript to parse the form data and put into the appropriate business logic implementations.

(5) The dynamic method invocation:

URL calls according to the rules in the same business logic object in the specified method.

Key processes are:

1. Client sends a request to the server, services send the client's request to the Filter.
2. Through analysis, filter and process of the requested data, and then passes the request to the next level of routing the dispenser.
3. Routing the dispenser according to the related configuration to determine the requested property is a business data or static resource request, pass to appropriate handlers.
4. If read as a static resource then requests a static resource returned to the client.
5. If business data request, depending on routing rules for forwarding to the appropriate business logic processing.
6. When processing is complete turn the data into JSON format back to the client.

see graph 5.1 below:

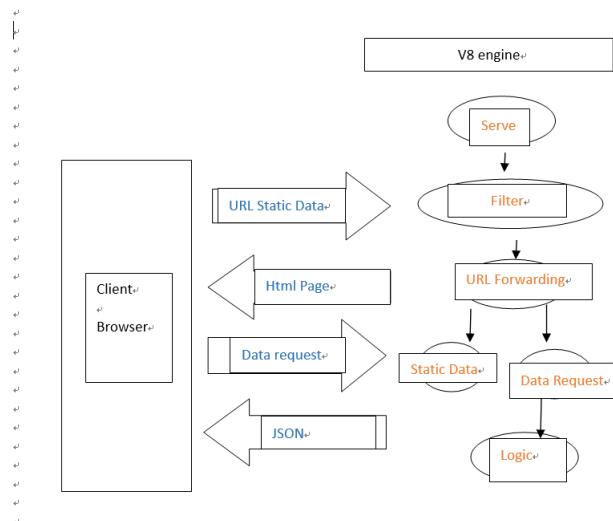


Figure 5.1: Key processing

5.4 System Functions Review(uml)

According to the functional requirements of the system, online shopping system is divided into the front-end management and back-end management. Front-end management, including browsing, query items, order merchandise, shopping carts, user information maintenance functions. Back-end management includes order management, items management, and user management module.

Front-end is described as follows:

1. Browse products

Product details

Item id

2 . Search for commodity

Product category

Product keyword

Order inquiries

3. Order Item

4. Shopping cart

5. Users information maintenance

User registration

User login

Modify user information

Back-end and described as follows:

1. Products management

Add a product category

Modify the product category

Delete product categories

Adding product information, including product category, name, number, company and other information;

Product image upload, modify, and delete;

Modify product information

Delete product information

View product information

2. Order management

Processing orders

Handling shipments

Checks out

Delete orders

3. Complaint management

Into complaints solutions

Delete a resolved complaints

View complaint support

4. Customer support management

Registered customer, including the user name, password and other information

Modify the customer information

Delete the customer information

5. User management functions of the system

Add system users, including user name, password and other information;
 Modify user information systems;
 Delete system user information.

And show a graph of those connection below graph 5.2

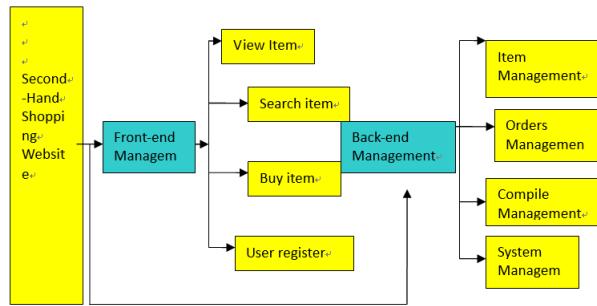


Figure 5.2: Website Function modular

Functions on Shopping website

In this System, user Management module is pretty simply. In this system have a default user is admin (username: Admin, Password: Admin) which can add consumer user to Database and do some operation such as delete, update and create. And Normal consumers user only can manage them own account modify name and password, and change items that post by themselves.

And show graph below graph 5.3:

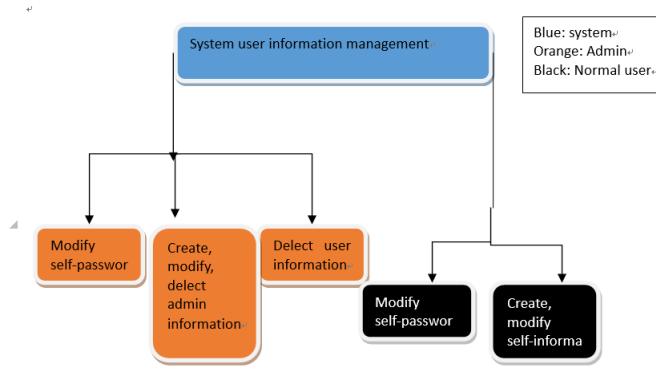


Figure 5.3: User Management

Normal Consumer shopping Steps show below graph 5.4:

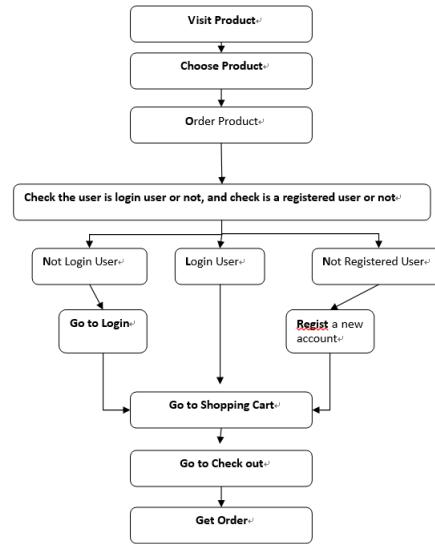


Figure 5.4: Normal User Shopping Steps

5.5 Security System

There have some special design I did for built a prefect security system.

5.5.1 Consumer's Password

When user register a new account, back-end will check this value of password.

(1) Check then length of password see graph 5.5a:

(2) When user click sign up back-end will convert consumers' password to a log string use Crypto modular. The algorithm it used is AES-128-ECB, and then use BASE64 to encoding. It make sure user's information will store in server-end safety. The graph 5.6 show below is encrypted password:

(a) Register page

(b) Password Check

(c) Password Check

Detailed description: The figure consists of three screenshots of a web application.
 (a) Register page: Shows a 'SIGN UP' form with fields for 'YOUR USERNAME', 'YOUR EMAIL', 'YOUR PASSWORD', and 'RETYPE PASSWORD'. Below the form is a 'Sign Up' button and a link 'ALREADY A MEMBER? GO AND LOGIN!'.
 (b) Password Check: Shows a message 'FORMAT ERROR' next to a field containing '.....'.
 (c) Password Check: Shows a message 'OK' next to a field containing a long, encoded password string.

Figure 5.5: Register page Password Check

```
Express server listening on port 3000
Original cleartext: wang123
base64 ciphertext: cNNyi6kQyLCjnFshmSjvWA==
UTF8 plaintext deciphered: wang123
```

Figure 5.6: Encryption

5.5.2 Session and Cookies

[4]

Cookies, session are method used to store user state information. The main difference is:

- (1) location: cookies are stored on the client side, the session on the server
- (2) Security: cookies, poor safety, the session high safety
- (3) life cycle: in the case of not set conditions both disappeared in the browser is turned off
(can be setting cookies on the client survival time, also can be installed on the server session time of survival)

Two relations, the session is implemented through a cookie

So, in my final project I choose use session to store the most data rather then use cookies

5.5.3 Database Security Design

In MongoDB 2.4 version it do a new adjustment on users right management, refine the permissions, enhanced security. So in this Project I also use Auth function create many different account to control different Database. If use an unauthorized account to login database, It's not allow to access into and can't get any data from database, and protect users and items information from server-end.

Open auth function

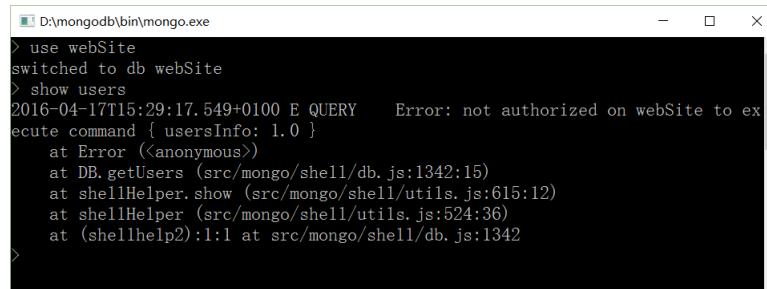
mongod --dbpath (db address) --port 27017 --logpath (mongodb.log address) --logappend --auth

Images 5.7&5.8 below is show the user and how it work: If use unauthorized account login



```
D:\mongodb\bin\mongo.exe
> db.auth('eee', 'eee');
1
> use admin
switched to db admin
> show collections
system.indexes
system.users
system.version
users
> db.users.find()
Error: error: { "$err" : "not authorized for query on admin.users", "code" : 1
3 }
>
```

Figure 5.7: Database account



D:\mongodb\bin\mongo.exe
> use webSite
switched to db webSite
> show users
2016-04-17T15:29:17.549+0100 E QUERY Error: not authorized on webSite to execute command { usersInfo: 1.0 }
at Error ((anonymous))
at DB.getUsers (src/mongo/shell/db.js:1342:15)
at shellHelper.show (src/mongo/shell/utils.js:615:12)
at shellHelper (src/mongo/shell/utils.js:524:36)
at (shellhelp2):1:1 at src/mongo/shell/db.js:1342

Figure 5.8: Reject access

5.5.4 Face to Face Trade

If consumer want to buy something from Second-Hand shopping website, they can use city-wide search get the seller's contact method such as email or phone number or Facebook and trade face to face.

Because consumers can't see the real situation of this product when they are shopping online. So, if they can trade with seller face to face they can check the product first and thinking buy or not.

5.6 Database design

In this project, I use MongoDB to store, update and delete user and item information.

5.6.1 Database Users

All users in Database is in table 5.1 below :

Database	Username	Password	Role
Admin	Harry	Harry	userAdminAnyDatabase
Admin	admin	admin	dbAdmin
webSite and webSite-app	normalUser	normalUser	readWrite
website-app	ccc	ccc	readWrite
website	eee	eee	readWrite

Table 5.1: Users

5.6.2 DB:webSite

(1)Collection: items

Items Table used to store all item's details such as color, name, id, price see table 5.2 below:

Property	Type
id	Number
username	String
location	String
category	String
picture	String
name	String
item_code	String
color	String
searchtimes	Number
number	Number
createTime	String
phone	Number
description	String
price	Number

Table 5.2: Items table

(2)Collection:Users

Users Table used to store all users' details such as picture, name, id, address see table 5.3 below:

Property	Type
id	Number
name	String
picture	String
password	String
email	String
phoneNumber	String
Address	String
flag	String

Table 5.3: Users table

(3)Shopping Cart table

Use graph 5.9 to store Cart information:

```

var name=docs[0].name;
var ord={id:req.body._id,
         name:docs[0].name,
         price:parseInt(docs[0].price),
         number:1,
         username:docs[0].username,
         category:docs[0].category,
         color:docs[0].color,
         picture:docs[0].picture
       };
  
```

Figure 5.9: Shopping Cart table

5.6.3 DB:webSite-app

Collections:sessions

This Database used to store Session data which can check login, realize shopping cart.see table 5.10 below:

D:\mongodb\bin\mongo.exe
> show dbs
admin 0.078GB
diamond 0.078GB
local 0.078GB
my_database_name 0.078GB
schools 0.078GB
test 0.078GB
webSite 0.078GB
webSite-app 0.078GB
> use webSite-app
switched to db webSite-app
> show collections
sessions
system.indexes
>

Figure 5.10: Session

5.6.4 webSite : Orders

This collection used to store all orders generated from website. see table 5.4

Property	Type
userid	Number
username	String
itemid	Array
name	Array
Address	String
createTime	String
total	Number

Table 5.4: Users table

5.6.5 Code-create tables

(1)Items Table

See graph 5.11:

(2)Users Table

See graph 5.12:

(3)Orders Table

```

1 var mongoose = require('mongoose');
2 var db=require("./user").db;
3 //连接数据库
4 console.log("item00 connected");
5 var Schema = mongoose.Schema; // 创建模型
6 var userschema = new Schema({
7   idNumber,
8   username : String,
9   location : String,
10  category : String,
11  picture : String,
12  name : String,
13  item_code : String,
14  color : String,
15  searchTime : String,
16  number : Number,
17  createTime : String,
18  phone : Number,
19  description : String,
20  price : Number
21 });
22 // 定义了一个新的模型，但是此模式还未和users集合有关联
23
24
25
26 exports.mongoose = mongoose;
27 exports.user = db.model('items', userSchema);
28 exports.userschema=userSchema;

```

Figure 5.11: Create Items code

```

1 var mongoose = require('mongoose');
2
3 //user : eee password : eee
4
5 var db = mongoose.connect('mongodb://eee:eee@localhost/webSite');
6 var Schema = mongoose.Schema; // 创建模型
7 var userschema = new Schema({
8   picture : String,
9   name : String,
10  password : String,
11  email : String,
12  phoneNumber : String,
13  Address : String,
14  flag : String
15 });
16 // 定义了一个新的模型，但是此模式还未和users集合有关联
17 if(db!=null)
18   console.log("Dbconnect");
19
20 exports.mongoose = mongoose;
21 exports.user = db.model('users', userSchema);
22 exports.db=db;
23 exports.userschema=userSchema;

```

Figure 5.12: Create User code

See graph 5.13:

```

1 var mongoose = require('mongoose');
2 var db=require("./user").db;
3 //连接数据库
4 console.log("item00 connected");
5 var Schema = mongoose.Schema; // 创建模型
6 var userschema = new Schema({
7   idNumberString,
8   username : String,
9   name: Array,
10  itemid : Array,
11  total : Number,
12  createTime : String,
13  Address : String
14 });
15 // 定义了一个新的模型，但是此模式还未和users集合有关联
16
17
18
19 exports.mongoose = mongoose;
20 exports.user = db.model('orders', userSchema);
21 exports.userschema=userSchema;

```

Figure 5.13: store orders

5.7 Core Function

5.7.1 Hottest Item

In the Main page of websit, the top of website have an exhibition window of rolling photographs. Every items in this window are hottest item recently, this rank list through searchtimes to decide. If there have a consumer visit a item, the searchtimes of this item will be add one. please see graph 5.14:

```
router.get('/item_detail_page/productname/:_id', function(req, res) {
    console.log(req.params._id);
    var cc;
    item.find({_id:req.params._id},function(err,docs){
        req.session.product=docs;
        console.log(docs);
        cc=docs;
        var vv=parseInt(docs[0].searchtimes)+1;
        console.log(parseInt(docs[0].searchtimes)+1);
        var conditions = {_id : req.params._id};
        var update = {$set : {searchtimes :vv}};
        var options = {upsert : true};
        item.update(conditions, update, options, function(error){
            if (error) {
                //code
            }else{
                console.log("success");
            }
        });
        res.render('item_detail_page', { title: 'item_detail_page' ,
            product:req.session.product,
            userpicture:req.session.userpicture,
            username:req.body.username,
            sum:req.session.sum
        });
    });
});
```

Figure 5.14: Function searchtimes adding

5.7.2 Search Function

In this website people can search item through location. It can help buyer find the nearest place to buy the product, and it can avoid fake item or long shipping time. Search function is below :



Figure 5.15: Location Search Function

5.7.3 Session

[4] This modular help project realize user keep login until user close browser. And it more security than cookies that store request data on link. Use Session also finish shopping cart function, when consumer add item to cart, store it on session. see graph 5.16a,5.16b to check some application on website:

5.8 Appearance Design

Following section will introduce my shopping website's interface design, it will be separated four parts.

5.8.1 Mainpage

This Section used to describe Main page design:

(1) Screen shot of main page see graph 5.17:



Figure 5.16: Register page Password Check

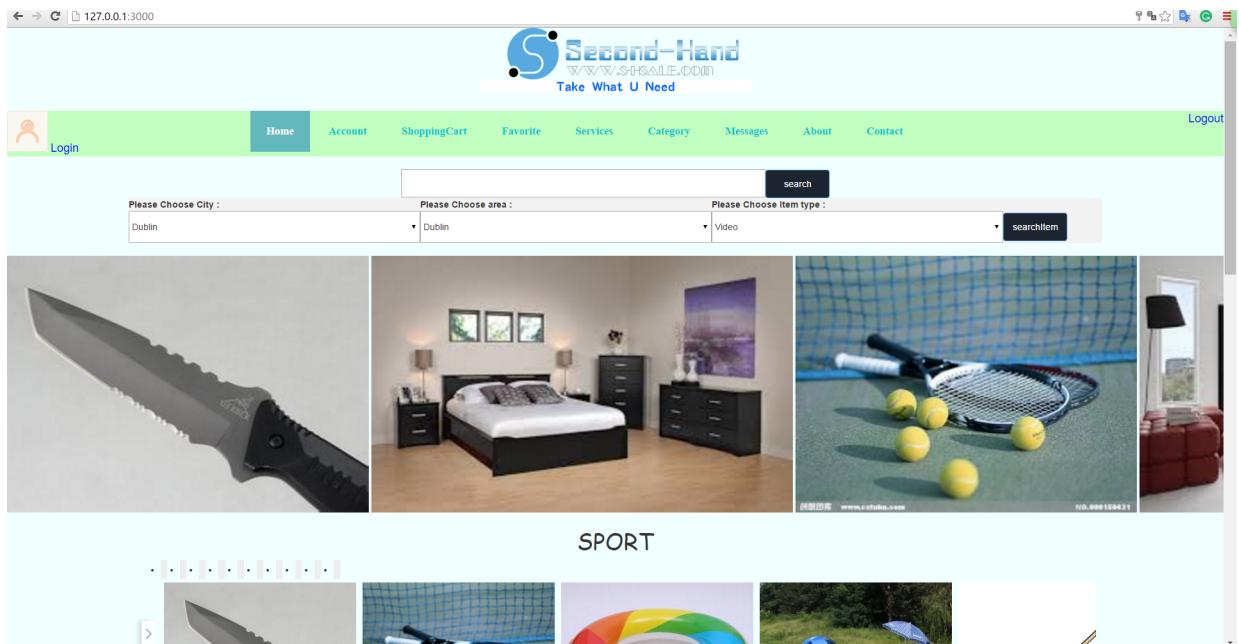


Figure 5.17: Main-page

(2)Items Exhibition Bar see graph 5.18:

In the left of Bar have two a arrow button which can switch to next page of items.

(3)Single Item View see graph 5.19:

1.When consumer move mouse top item will show picture like this.

2.The green button that in the centre of graph, when it be pressed will jump to item_detail page.

3.In the bottle of graph is the name and price of product, and there also have a Pay-Pal Buy-Now Button help client to quick shopping.



Figure 5.18: Items Exhibition Bar

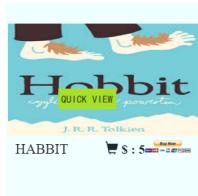


Figure 5.19: Single-item

5.8.2 User Management Page

(1) Login Page see graph 5.20

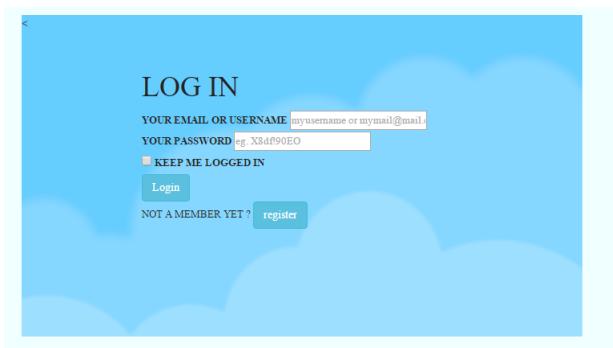


Figure 5.20: Login

(2) Register Page see graph 5.21 This page require consumer fill table to register an account for shopping website.

(3) User Information Page see graph 5.22:

In the left of graph have 5 links which can realize different functions such as update personal information, sell product and manage selling product.

(4) Information update Page see graph 5.23:

Clients can Change his personal information such as Picture, Phone Number, Address.

(5) All Orders see graph 5.24:

(6) Admin Page see graph 5.25:



Figure 5.21: Register



Figure 5.22: User Information Page

Admin Page only allow access by Administrator which username is "admin" and password is "admin".

Administrator can update and delete users in website.

In new version, people can not use same email to register a new account.

5.8.3 Items Page

(1) Add Product that you want to sale see graph 5.26:

(2) Selling Products Management see graph 5.27: This Page below show all items that be created by consumers. Include some details such as name, price, createTime.

(3) Items Detail Page see graph 5.28: In the bottle of Graph is an area offer consumer write down them felling and feedback about this product.

(4) OrderCheck Page see graph 5.29: First area is to set delivery address.

Second is check and confirm order is or not correct.

Then click PayNow button.

(5) Search Page see graph 5.30: All products will be ranked by search times.

(6) Shopping Cart see graph 5.31:

(7) Shopping Cart Mini State window see graph 5.32:

Second-Hand Shopping Website Report

The screenshot shows a form titled "Modify - Necessary Content". It includes fields for "Username" (admin), "Picture" (a placeholder image), "Phone Number" (未设置任何信息), and "Delivery Address" (a large empty text area). On the left, there is a sidebar with a list of options: *Admin, *Personal Information Update, *membership, *Add product, and *all selling product.

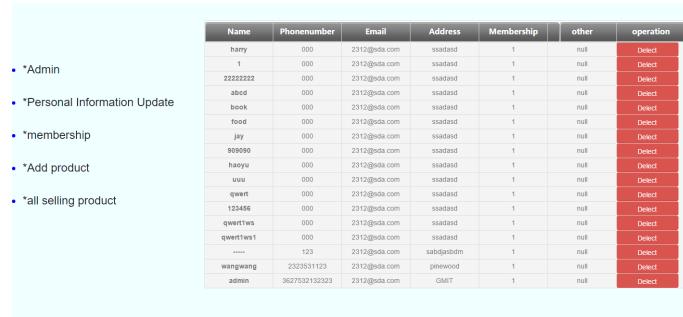
Figure 5.23: Information update Page

The screenshot shows a navigation menu at the top with links: Home, Account, ShoppingCart, Favorite, Services, Category, Logout, Messages, About, and Contact. Below the menu, there is a sidebar with the same list of options as Figure 5.23. The main content area displays a table of "All Orders".

Name	CreateTime	Total Price	operation
admin	24-04-2016 02:10:07	\$.333	Delete
admin	24-04-2016 02:13:18	\$.720	Delete

Figure 5.24: All orders Page

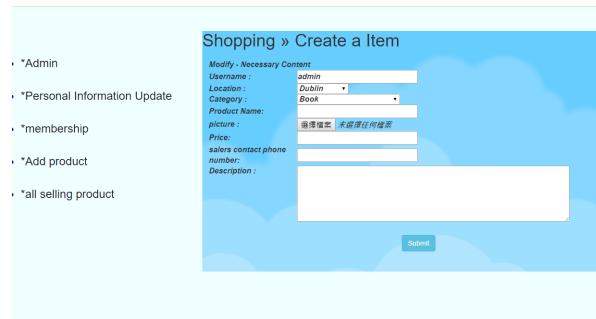
When consumers put mouse top Cart State will show this.



The screenshot shows a table with columns: Name, PhoneNumber, Email, Address, Membership, other, and operation. The data includes various users like harry, 1, 22222222, abcd, book, food, jay, 909090, haoyu, uuu, quart, 123456, quetties, quarties1, ----, wangwang, and admin, each with their respective details and actions.

	Name	PhoneNumber	Email	Address	Membership	other	operation
• *Admin	harry	000	2312@qoda.com	ssaaasd	1	null	Delete
• *Personal Information Update	1	000	2312@qoda.com	ssaaasd	1	null	Delete
• *membership	22222222	000	2312@qoda.com	ssaaasd	1	null	Delete
• *Add product	abcd	000	2312@qoda.com	ssaaasd	1	null	Delete
• *all selling product	book	000	2312@qoda.com	ssaaasd	1	null	Delete
	food	000	2312@qoda.com	ssaaasd	1	null	Delete
	jay	000	2312@qoda.com	ssaaasd	1	null	Delete
	909090	000	2312@qoda.com	ssaaasd	1	null	Delete
	haoyu	000	2312@qoda.com	ssaaasd	1	null	Delete
	uuu	000	2312@qoda.com	ssaaasd	1	null	Delete
	quart	000	2312@qoda.com	ssaaasd	1	null	Delete
	123456	000	2312@qoda.com	ssaaasd	1	null	Delete
	quetties	000	2312@qoda.com	ssaaasd	1	null	Delete
	quarties1	000	2312@qoda.com	ssaaasd	1	null	Delete
	----	123	2312@qoda.com	subqodbdm	1	null	Delete
	wangwang	2323531123	2312@qoda.com	pniewood	1	null	Delete
	admin	3627532152323	2312@qoda.com	GMIT	1	null	Delete

Figure 5.25: Selling Products Management



The screenshot shows a 'Create a Item' form with fields for Username (admin), Location (Dublin), Category (Book), Product Name, picture (a placeholder image), Price (1), and Description (未设置任何描述). There is also a 'Modify - Necessary Content' section with a 'Submit' button.

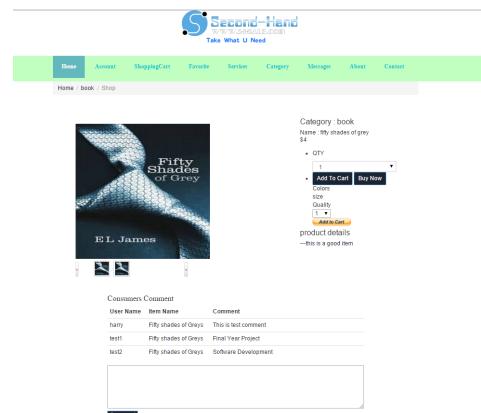
Figure 5.26: Add Selling Products



The screenshot shows a table with columns: Name, Price, Description, Time, Number, Category, Update, and Delete. It lists two products: one with ID 2221 (Price 1, Description Testing, Time 2016-04-22 00:47:14, Number 12, Category Toys) and another with ID 232 (Price 324, Description Testing, Time 2016-04-22 00:42:47, Number 3, Category Handmade). There are 'UPDATE' and 'Delete' buttons for each row.

	Name	Price	Description	Time	Number	Category	Update	Delete
	2221	1	Testing	2016-04-22 00:47:14	12	Toys	<button>UPDATE</button>	<button>Delete</button>
	232	324	Testing	2016-04-22 00:42:47	3	Handmade	<button>UPDATE</button>	<button>Delete</button>

Figure 5.27: Selling Products Management



The screenshot shows the product detail page for 'Fifty Shades of Grey'. It features the book cover, a summary (Name: Fifty shades of grey, Author: E.L. James), and purchase options (Qty: 1, Add To Cart, Buy Now). Below the summary, there's a 'product details' section with a note: 'This is a good item'. At the bottom, there's a 'Comments' section with a table of consumer comments from users harry, test1, and test2.

User Name	Item Name	Comment
harry	Fifty shades of Grey	This is test comment
test1	Fifty shades of Grey	Final Year Project
test2	Fifty shades of Grey	Software Development

Figure 5.28: Item Detail Page

Second-Hand Shopping Website Report

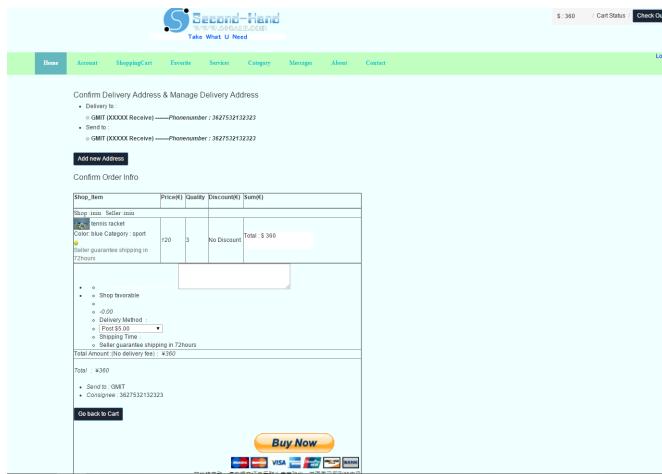


Figure 5.29: OrderCheck Page

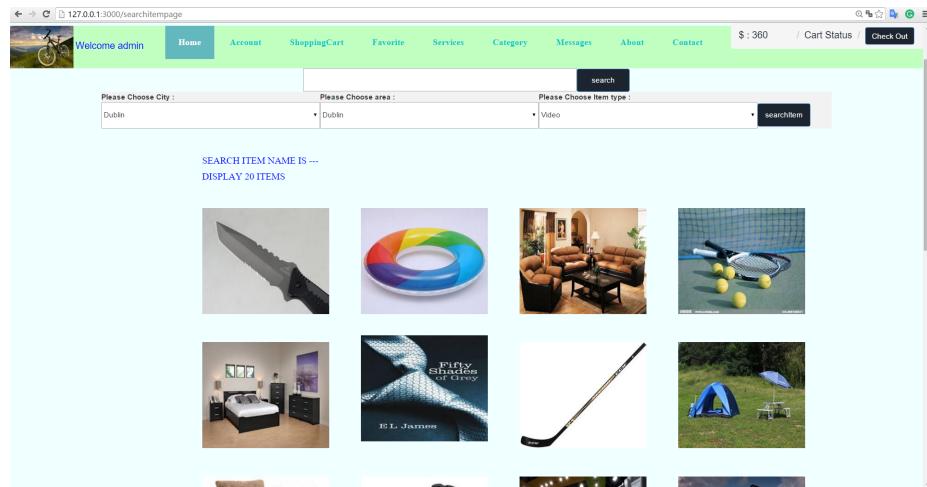
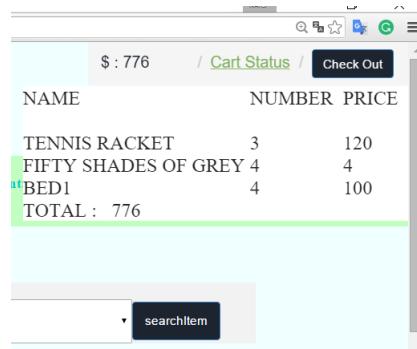


Figure 5.30: Search Page



Figure 5.31: Shopping Cart



A screenshot of a shopping cart status window. The window has a title bar with icons for search, refresh, and other functions. Below the title bar, the text '\$: 776' is displayed, followed by a separator line and the text 'Cart Status / Check Out'. The main content area is a table with three columns: NAME, NUMBER, and PRICE. The table lists three items: 'TENNIS RACKET' (NUMBER 3, PRICE 120), 'FIFTY SHADES OF GREY' (NUMBER 4, PRICE 4), and 'BED1' (NUMBER 4, PRICE 100). A green horizontal bar at the bottom of the table area contains the text 'TOTAL : 776'. At the bottom of the window, there is a dark button labeled 'searchItem'.

NAME	NUMBER	PRICE
TENNIS RACKET	3	120
FIFTY SHADES OF GREY	4	4
BED1	4	100
TOTAL : 776		

Figure 5.32: Shopping Cary Mini State window

Chapter 6

System Evaluation

6.1 Advantages

This is a Nodejs project which have better performance than JSP, it can run and get data form back-end faster. And in website application i design many part to ensure protect consumers' information such as encrypt password, authorised database and used session.

6.2 Limitation

Due to the short development time, website design is not good enough, and there also have a lot of flaws in code. But I think this website will better with the Node.js platform developing and continue maintains.

Code inefficiency.

Firstly, manage and control node.js asynchronous processing mechanism not skilled enough cause short development time. Typing a lot of code that inefficient. If the requested data objects in the data object has too many levels may lead to response time too long to respond to interrupts

Function realization not rigorous.

Certain features in the website are not completed finished because the time problem. When consumer frequent refresh page or visit this website maybe will result in crashes. For example Session data. Not strictly limit each user can hold Session data size. If there is a storing data in session , because the Session data has been stored in memory, and so easily because the data is too large causing memory overflow problem.

Log function is missing.

Node.js's platform log output to the console can print certain information. In the early time of website design do not have a completed log processing module, just printed on the console. With the application of the project, once the error occurred during the operation, just displayed in the console does not work once the console closes these error messages follow disappear. The server-end requires a logging module, framework error need be logged in the log file, help to exclude errors. Currently this feature is not exist yet.

Didn't bind a domain to website, it's a biggest problem.

The content above is talking about the limitations of website, and following introduction is describe future development of the project and functions issues;

6.3 Future Development and Optimization

Optimize my code.

Optimize some code for functions, using a high efficient algorithms to do some data processing.

Implementing logging.

Finish logging module to store Log error messages to a local memory , you can configure the output format and the output according to the information level.

Bind a domain to website, it can provide a better experience to consumer and also can test help people to buy low-price product or dispose furniture or electric applications they don't need.

The website source code will host on GitHub so that I can improve and modify, make it more prefect and efficient. Open over the network can be modified together looking for people who possess the same interest so people know and are willing to try. This is the ultimate purpose of this framework.

This chapter introduced the framework of the specific deficiencies and inadequate quarters do revise and perfect plan. And illustrate the development of the framework for the final desired result.

Chapter 7

Conclusion

After nearly two months of unremitting efforts, finally finished my graduation website Design. The Application's title is "Second-Hand shopping website".

When i first time know final year design is no limits on type, im so excited because i can do the Second-hand shopping website that raised this idea when i arrive Ireland. At begin of project I went to the library, bookstore to check a lot of information on network database about the "online shopping system".

This Project was made by the Nodejs, database was built by MongoDB. To that end, I took three weeks to get used to write Nodejs page and the MongoDB database using the method. Through the above preparatory work, made my follow-up developed smoothly, will not affect the progress of some difficulty, guarantee paper finish at time.

When i begin designing this system, primarily defining the database and initialize the data. This technical work is not strong, simply enter all data. When database design is complete, and start entering the Web page programming, is the most time consuming job. In this process, the problems encountered are mostly grammatical problems, also had a few appearances database connection. Such as database storage path is changed while in use, leads to cannot access the data via a Web page, only to specify a path statement in your Web page code make the appropriate modifications, to make it accessible.

In the system design process, I got help from classmates, they make my schedule did not stagnate, let me save a lot of detours. It also made me see the truth, no one can execute, do it all without missing, something beyond the reach of your place. Communication is the best way to solve these problems. Through this graduation thesis design, not only improved my ability to solve problems, is the exercise of my own programming ability, I provided valuable experience for future development.

Due to time constraints and the lack of practical experience, and performed by the graduation system where many have yet to be improved.

Overall, graduated design although end has, by prepared of software although also has many of vulnerability, believes in teacher's eyes this project also has many child-

ish and insufficient place, but this graduated design does up to has is big of effect, it is a bog check by learn to of knowledge in this year, also let me have new awareness to future career. I think the results of the design is not important, summed up the most important thing is the experience and knowledge they acquired from the production process.

Appendix A

Appendix

<https://github.com/G00330443/Finalyear-Project.git>

All available source in Github link :

<https://github.com/G00330443/Finalyear-Project.git>

Have some problems when you click the link, please careful see the end of address always have [numbers].

Please delect it then you can visit the correct address, sorry about that.

Learning resource:

Some base information of nodejs, such as how to set environment and data type

<http://www.tutorialspoint.com/nodejs/>

Youtube tutorial

<https://www.youtube.com/channel/UCJbPGzawDH1njbqV-D5HqKw>

Nodejs with mongodb

<https://www.youtube.com/watch?v=CIPbmPUKyMI>

<https://www.youtube.com/watch?v=ZKwrOXI5TDI>

Nodejs send email use Nodemailer

<https://nodemailer.com/>

A Blog website

<http://blog.fens.me/nodejs-express3/>

<https://github.com/bsspirit/nodejs-demo>

website Design: Amazon and e bay website.

Bibliography

- [1] AJAX. *AJAX*. -. URL: https://en.wikipedia.org/wiki/Ajax_%28programming%29.
- [2] Implusive Buying. *The Psychology of Impulse Buying*. 2005. URL: <http://specialtyretail.com/issue/2005/10/merchandising-and-marketing/the-psychology-of-impulse-buying/>.
- [3] Nicholas C.Zakas. *jj JavaScript advanced program designżż*. Reading, JavaScript: —, 2005.
- [4] Github. *Session*. -.
- [5] Jeff Harrell. *Node.js at PayPal*. -. URL: <https://wwwpaypal-engineering.com/2013/11/22/node-js-at-paypal/>.
- [6] JSON. *JSON*. -. URL: <https://en.wikipedia.org/wiki/JSON%20;%20http://www.copterlabs.com/json-what-it-is-how-it-works-how-to-use-it/>.
- [7] David Mielach. *Can't Resist That Gum? You're Not Alone*. -. URL: <http://www.businessnewsdaily.com/2370-impulse-purcahse-survey.html>.
- [8] Mongoose. *Mongoose*. -. URL: <http://mongoosejs.com/docs/api.html>.
- [9] Stacey Rudolph. *The Future of Online Retail Shopping is Bright—Statistics and Trends*. February 18,2016. URL: <http://www.business2community.com/infographics/future-online-retail-shopping-bright-statistics-trends-01458175#8EsZoJP4E27AGvdq.97>.
- [10] TKIRDA. *autocomplete*. -. URL: <https://github.com/devbridge/jQuery-Autocomplete>.
- [11] Wiki. *Github*. -. URL: <https://en.wikipedia.org/wiki/GitHub>.
- [12] Wiki. *KomodoEdit*. -. URL: https://en.wikipedia.org/wiki/Komodo_Edit.
- [13] Wiki. *MongoDB*. -. URL: <https://en.wikipedia.org/wiki/MongoDB>.
- [14] Wiki. *Node.js*. -. URL: <https://en.wikipedia.org/wiki/Node.js>.
- [15] Eames Yates. *The biggest security mistakes people make when buying things online*. 2015. URL: <http://uk.businessinsider.com/online-shopping-security-risks-internet-malware-adam-levin-2015-11?r=US&IR=T>.