**HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY OF INFORMATION TECHNOLOGY**

Logo

Description automatically generated

**Course name: IT Project**

**SNEAKER SHOP WEBSITE**

**Lecturer name**: Mai Anh Thơ

**List of members**

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student name** | **Contribution (%)** |
| 20110352 | Nguyễn Việt Anh | 100% |
| 20110403 | Lê Y Thiện | 100% |

*Ho Chi Minh City, December 2022*

Contents

[PREFACE 4](#_Toc121334892)

[I. Specific description 5](#_Toc121334893)

[1. Software purpose: 5](#_Toc121334894)

[2. Input data: 5](#_Toc121334895)

[3. Use case: 5](#_Toc121334896)

[4. Expected interface: 6](#_Toc121334897)

[II. Work distribution: 6](#_Toc121334898)

[III. Design 7](#_Toc121334899)

[1. Algorithms: 7](#_Toc121334900)

[2. Classes: 8](#_Toc121334901)

[A. All classes 8](#_Toc121334902)

[B. All Methods: 11](#_Toc121334903)

[3. Database Desgin 18](#_Toc121334904)

[4. Fields of each table: 19](#_Toc121334905)

[5. Interface design 21](#_Toc121334906)

[IV. Deploying and Testing: 28](#_Toc121334907)

[Scenario 1: Signup/Log in before conduct buying. 28](#_Toc121334908)

[Scenario 2: Conduct shoping 29](#_Toc121334909)

[Scenario 3: Conduct shoping without signed in 30](#_Toc121334910)

[Scenario 4: Admin sign in to 31](#_Toc121334911)

[V. Conclusion 32](#_Toc121334912)

[Pros: 32](#_Toc121334913)

[Cons: 32](#_Toc121334914)

[Future development: 32](#_Toc121334915)

[Link source code: 33](#_Toc121334916)

[Reference Source: 33](#_Toc121334917)

**List of Figure**

[**Figure 1 MVC model** 7](#_Toc121321900)

[**Figure 2 Database** 18](#_Toc121321901)

**List of Table**

[**Table 1 Use case** 5](#_Toc121321993)

[**Table 2 Interface** 6](#_Toc121321994)

[**Table 3 Work Distribution** 6](#_Toc121321995)

[**Table 4 Author's class: Thien & Anh** 8](#_Toc121321996)

[**Table 5 Author's class: Nguyen Viet Anh** 9](#_Toc121321997)

[**Table 6 Author’s Class: Le Y Thien** 10](#_Toc121321998)

[**Table 7 Author's Method: Nguyen Viet Anh** 11](#_Toc121321999)

[**Table 8 Author's Method: Le Y Thien** 16](#_Toc121322000)

[**Table 9 List of table of database** 18](#_Toc121322001)

[**Table 10 Field of User** 19](#_Toc121322002)

[**Table 11 Field of Product** 19](#_Toc121322003)

[**Table 12 Field of Invoice** 20](#_Toc121322004)

[**Table 13 Field of Detail\_In** 20](#_Toc121322005)

[**Table 14 Interface Design** 21](#_Toc121322006)

[**Table 15 Scenario 1** 28](#_Toc121322007)

[**Table 16 Scenario 2** 29](#_Toc121322008)

[**Table 17 Scenario 3** 30](#_Toc121322009)

[**Table 18 Scenario 4** 31](#_Toc121322010)

# PREFACE

The first words allow the author to express sincere and deep gratitude to Mai Anh Thơ - the scientific instructor for the author, directed the topic, directly guided and had valuable suggestions to help the author during the research implementation. The author would like to thank the teachers and teachers of Ho Chi Minh City University of Technology and Education for teaching and imparting valuable knowledge to the author as the foundation for this study.

At the same time, the author would like to thank the Masters and Postgraduate Department of Ho Chi Minh City University of Technical Education for their enthusiastic help and guidance in completing the procedures related to the study. assist. The author also would like to sincerely thank you, sisters and colleagues for creating favorable conditions, providing documents, discussions to help the author complete the scales, surveys and take time to refer. and answer the survey.

Finally, the author would like to thank family and friends for their help and for creating a solid support for the author during the study and to obtain the research results.

Sincerely thanks.

Ho Chi Minh City, December 2022 Authors

Nguyễn Việt Anh

Lê Y Thiện

# I. Specific description

## 1. Software purpose:

A responsive shopping website that allows customers to buy sneakers, shoes. Apart from that, administrator can manage the list of products on the website.

## 2. Input data:

We use Clear Database MySQL to stores list of products, Invoice and Users.

Request from website.

## 3. Use case:

**Table 1 Use case**

|  |  |  |
| --- | --- | --- |
| **Order** | **Use case** | **Description** |
| **1** | Login/ Logout | Sign in/Sign out to use the website. |
| **2** | Register | Create a new user and store information to database.  Send an authentication code to user email to verify account. |
| **3** | Manage Product | Add, Delete, Edit a specific product by administrator. |
| **4** | Manage Invoice | Control the list of products which user has bought.  Store customer invoices to the database |
| **5** | Order confirmation | Send a confirmation email contain bill to customer. |
| **6** | Cart control | Add product which user click buy to cart.  Allow user to increase/decrease, remove the amount of product from cart. |
| **7** | Search product | Allow customers to search a specific product on shopping page. |
| **8** | Filter product | Allow user to sort product by brands. |
| **9** | Show product detail | Display product information. |

## 4. Expected interface:

**Table 2 Interface**

|  |  |
| --- | --- |
| **Page** | **Purpose** |
| Homepage | Shop homepage. |
| About page | Contains shop information. |
| Shopping page | List all product available |
| Check out page | Cart and costumer’s information for payment. |
| Admin page | Allow administrator to manage products. |
| Login/Logout, Register page | Allow user to sign in, sign out and register to the website. |
| Buy history page | Display all product that user has bought. |

# II. Work distribution:

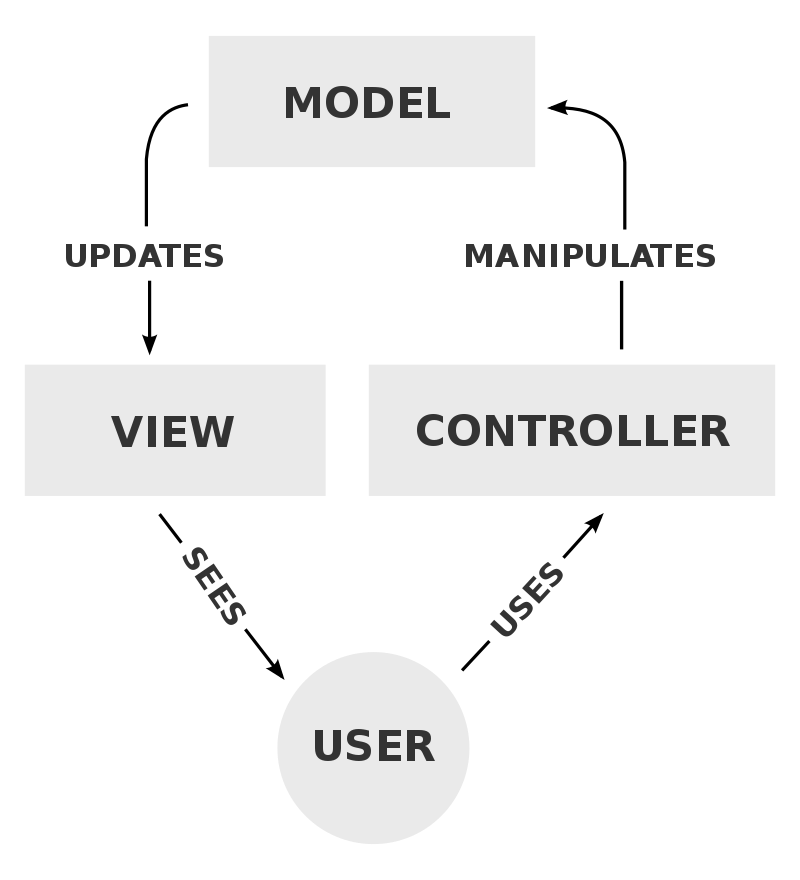
**Table 3 Work Distribution**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Work detail** | **Contribution** |
| 20110352 | Nguyễn Việt Anh | * Connect to database:   + Load data: list of products.   + Search product.   + Filter product.   + User buy history. * Send email: verify code, bill confirmation. * Admin: add, delete, edit product. * Front-end customize. | 100% |
| 20110403 | Lê Y Thiện | * Login/Logout, Register using session/cookie. * Cart control using session/cookie.   + Add, remove product to cart.   + Process amount of product in cart. * Store invoice to database. * Front-end customize. | 100% |

# III. Design

## 1. Algorithms:

Our website build base on Model–view–controller (MVC) architecture. That divide the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.



**Figure 1 MVC model**

We use jdbc to connect to database use Clear Database MySQL.

Process request: post and get method.

## 2. Classes:

### A. All classes

Author: Nguyễn Việt Anh 20110352 + Lê Y Thiện 20110403

**Table 4 Author's class: Thien & Anh**

|  |  |  |
| --- | --- | --- |
| **Order** | **Class name** | **Purpose** |
| **1** | ProductControl  (extends HttpServlet) | Annotation: /productload  Load all product to website. |
| **2** | LoadData | Execute query to load/store data from database for different other function |

Author: Nguyễn Việt Anh 20110352

**Table 5 Author's class: Nguyen Viet Anh**

|  |  |  |
| --- | --- | --- |
| **Order** | **Class name** | **Purpose** |
| 1 | DataBaseConnection | Connect to database. |
| 2 | User | Create User object to store user information and used further function. |
| 3 | Product | Create Product object to store user information and used further function. |
| 4 | Type | Create Type object to store type of object for filter function. |
| 5 | SendEmail | Send email: send code for verification and send bill for oreder confirmation. |
| 6 | AdminControl  (extends HttpServlet) | Annotation: /admincontrol  Load the list of products in the database. Applying paging function. |
| 7 | AddControl  (extends HttpServlet) | Annotation: /add  Add a new product to database. |
| 8 | DeleteControl  (extends HttpServlet) | Annotation: /adminDetail  Delete a product by ID from database. |
| 9 | AdminDetailControl (extends HttpServlet) | Annotation:  Load the product information by ID and direct to edit.jsp. |
| 10 | AdminEditControl (extends HttpServlet) | Annotation: /edit  Get the product information from edit.jsp page then edit product in database. |
| 11 | EmailVerifyControl (extends HttpServlet) | Annotation: /delete  Send a verification code to user email after filled in register.jsp. |
| 12 | VerifyCodeControl (extends HttpServlet) | Annotation: /verifycode  Check the verification code user input correct |
| 13 | ListProduct  (extends HttpServlet) | Annotation: /ListProduct  Load Product to different pages with index. |
| 14 | CatergoryControl  (extends HttpServlet) | Annotation: /category  Display all products of a specific brand. |
| 15 | DetailControl  (extends HttpServlet) | Annotation: /detail  Display all information of a specific product. |
| 16 | SearchControl  (extends HttpServlet) | Annotation: /search  Display all product with the name contains character user inputs at search bar. |
| 17 | UserHistoryControl  (extends HttpServlet) | Annotation: /userhistorycontrol  Display all product of user has bought. |

Author: Lê Y Thiện 20110403

**Table 6 Author’s Class: Le Y Thien**

|  |  |  |
| --- | --- | --- |
| **Order** | **Class name** | **Purpose** |
| 1 | LineItem | Create an object to store Product and the amount which was bought by user. |
| 2 | Cart | Create an object to store the list of LineItem which was bought buy customer |
| 3 | BeforeLogin  (extends HttpServlet) | Annotation: /beforelogin  Check the account if an account has been logged in recently |
| 4 | LoginControl  (extends HttpServlet) | Annotation:  Check the user if an account was registered or not |
| 5 | LogoutControl  (extends HttpServlet) | Annotation: /login  Exit current account |
| 6 | RegisterControl  (extends HttpServlet) | Annotation: /logout  Support for register an account |
| 7 | BuyControl  (extends HttpServlet) | Annotation: /buy  Support for buying items from website then add to cart |
| 8 | ShowCartControl  (extends HttpServlet) | Annotation: /checkout  Display a list of products and quantities purchased by  customer |
| 9 | ProcessAmount  (extends HttpServlet) | Annotation: /process  Supporting adjust quantities of Products purchased and  remove it |
| 10 | Payment  (extends HttpServlet) | Annotation: /payment  Store the bill to database and send email to confirm bill |

### B. All Methods:

Author: Nguyễn Việt Anh 20110352

**Table 7 Author's Method: Nguyen Viet Anh**

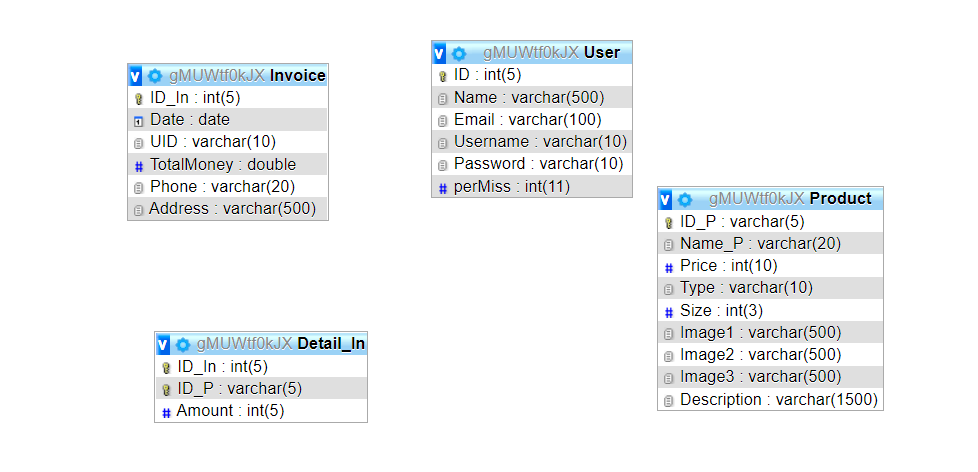
|  |  |  |  |
| --- | --- | --- | --- |
| **Order** | **Method** | **Purpose** | **FileName**  **(Line Contain Declaration)** |
| **1** | processRequest(request, response)  Input: get “index” at admin.jsp from request.  Output:  Push to request:  Index as “ activeIndex”.  endPage as “endP”.  listProductIndex as “adminListP”.  list as “adminListT”  Direct request to page: admin.jsp | * Load all product from database and apply Pagination. Each page will have 9 products. * Load the all the type of product. | **AdminControl.java (35)** |
| **2** | processRequest(request, response)  Input: get all product’s information from request.  Output: redirect request to admincontrol servlet | * Get id, name, image1, image2, image3, price, size. description, type from request * Insert to database a new product with the information above. | **AddControl.java (32)** |
| **3** | ProcessRequest(request, response)  Input: get the “delid” from request.  Output: redirect request to admincontrol servlet | * Get delid from request and conduct delete product which has that delid. | **DeleteControl.java (32)** |
| **4** | ProcessRequest(request, response)  Input: get the “pid” from request.  Output: Push to request product p as “detail”, list of types list as “ListT” and send request to Edit.jsp | * Get the information of product has pid get from request and also get list of types. | **AdminDetailControl (35)** |
| **5** | ProcessRequest(request, response)  Input:  Get all product information from request  Output: redirect to admincontrol servlet | * Edit product with the same id in database with the information get from request. | **AdminEditControl (32)** |
| **6** | processRequest(request, response)  Input: Get user information from request.  Output:  Push user to session and redirect to verify.jsp | * Get user information from request. * Generate a verification code * Create a new user with information and code above. * Send the code to the user email. * Push user to session | **EmailVerifyControl.java (33)** |
| **7** | processRequest(request, response)  Input: get user from session  Output:  Direct to verify.jsp if code is incorrect.  Direct to register servlet if code is correct. | * Get the user from session * Get authcode from request. * Compare authcode with user.getCode() | **VerifyCodeControl.java (33)** |
| **8** | processRequest(request, response)  Input: Output:  Push to request endPage as “endP” and direct to shop.jsp | * Calculate ending page for Pagination. | **ListProduct.java (32)** |
| **9** | processRequest(request, response)  Input:  Get all type from request.  Output:  Push to request:  list of products by type: list as “ListP”  list of types: list as “ListT”  3 latest products: last3P as “Last3p”  And direct to shop.jsp | * Load from database all product with type get from request. | **CategoryControl.java (35)** |
| **10** | processRequest(request, response)  Input: get pid from request  Output: push to request product: p as “detail” and direct to single.jsp | * Load product by ID get from request. | **DetailControl.java (33)** |
| **11** | processRequest(request, response)  Input: get search from request  Output: Push to request:  List of products by name: list as “ListP”  list of types: list as “ListT”  3 latest products: last3P as “Last3p”  And direct to shop.jsp | * Load product by name base on search bar. | **SearchControl.java (35)** |
| **12** | processRequest(request, response)  Input: get userid from request  Output: Push to request list of product that user has bought: list as “ListUH” and direct to buyhistory.jsp | * Load the list of product has bought of the user has userid get from request. | **UserHistoryControl.java (34)** |
| **13** | sendCode(User user)  Input: user  Output: true/false | * Send a email contain verify code to user.getEmail() | **SendEmail.java (33)** |
| **14** | sendBill(User user, String fullName, String phoneNumber, String address, String total)  Input: user, fullName, phoneNumber, address, total.  Output: true/false | * Send a bill to customer email to customer email contain: customer name, phonenumber, address and total money of that bill. | **SendEmail.java (72)** |

Author: Lê Y Thiện 20110403

**Table 8 Author's Method: Le Y Thien**

|  |  |  |  |
| --- | --- | --- | --- |
| **Order** | **Method** | **Purpose** | **FileName**  **(Line Contain Declaration)** |
| **1** | doPost (request, response)  Input: get user and password from request.  Output: Set username and password into request and redirect to LoginPage | Check the account if an account has been logged in recently | **BeforeLogin.java (23)** |
| **2** | doPost (request, response)  Input: get Username and Password from request  Output: Set an account to request and cookie and redirect to HomePage | +Check the account in database if an account has been registered or not.  +Check this account is customer or admin | **LoginControl.java (23)** |
| **3** | doPost (request, response)  Input: do not have  Output: Remove account from session/cookie and remove Cart of this account.  Redirect to HomePage | +Exit current account  +Renew cart on website  +Refresh cookie | **LogoutControl.java (27)** |
| **4** | doPost (request, response)  Input: get information from form in View  Output: Store account into database and set an account to session/cookie.  Redirect to HomePage | +Supporting user create an account for shopping | **RegisterControl.java (21)** |
| **5** | doPost (request, response)  Input: get Amount and Id of Product  Output: Store Product to Cart then set Cart into cookie. | +Supporting buy items by customer | **BuyControl.java (27)** |
| **6** | doPost (request, response)  Input: get list of products from cookie and store into Cart  Output: Set Cart into request.  Redirect to CheckoutPage | +Display the list of products which was bought by customer  +Show total money of customer’s Cart | **ShowCartControl (26)** |
| **7** | doGet (request, response)  Input: get product Id and signal increase or decrease product.  Output: Set Cart into cookie  +Refresh current page | +Supporting the change of product quantity | **ProcessAmount.java (80)** |
| **8** | doPost (request, response)  Input: get product Id  Output: Set new Cart into cookie  +Redirect to CheckoutPage | +Supporting for remove products | **ProcessAmount.java (36)** |

## 3. Database Desgin



**Figure 2 Database**

**Table 9 List of table of database**

|  |  |  |
| --- | --- | --- |
| **Order** | **Table name** | **Purpose** |
| **1** | User | Store user information  (name, email, username, password, permission) |
| **2** | Product | Store product information  (ID, name, price, type, size, images, descriptoin) |
| **3** | Invoice | Store Invoice of a user  (ID, date, userID, total money, Phone, Address) |
| **4** | Detail\_In | Store detail of a Invoice  (ID of invoice, ID of product, amount) |

## 4. Fields of each table:

Field of User table:

**Table 10 Field of User**

|  |  |  |  |
| --- | --- | --- | --- |
| Order | **Column name** | **Data type** | **Purpose** |
| **1** | ID | Int | Store user’s ID, auto increase |
| **2** | Name | varchar | Store user’s name |
| **3** | Email | varchar | Store user’s email |
| **4** | Username | varchar | Store user’s username |
| **5** | Password | varchar | Store user’s password |
| **6** | perMiss | int | Store user permission (admin or not) |

Field of Product table:

**Table 11 Field of Product**

|  |  |  |  |
| --- | --- | --- | --- |
| Order | **Column name** | **Data type** | **Purpose** |
| **1** | ID\_P | Varchar | Store product ID |
| **2** | Name\_P | varchar | Store product name |
| **3** | Price | varchar | Store product price |
| **4** | Type | Varchar | Store product type |
| **5** | Size | Int | Store product size |
| **6** | Image1 | Varchar | Store product first image |
| **7** | Image2 | Varchar | Store product second image |
| **8** | Image3 | Varchar | Store product third image |
| **9** | Description | Varchar | Store product description |

Field of Invoice table:

**Table 12 Field of Invoice**

|  |  |  |  |
| --- | --- | --- | --- |
| Order | **Column name** | **Data type** | **Purpose** |
| **1** | ID\_In | Int | Store invoice ID |
| **2** | Date | date | Store date of purchasing. |
| **3** | UID | varchar | Store user ID who purchased. |
| **4** | TotalMoney | Double | Store the total money |
| **5** | Phone | varchar | Store user phone |
| **6** | Address | Varchar | Store user address |

Field of Detail\_In table:

**Table 13 Field of Detail\_In**

|  |  |  |  |
| --- | --- | --- | --- |
| Order | **Column name** | **Data type** | **Purpose** |
| **1** | ID\_In | Int | Store the ID of Invoice |
| **2** | ID\_P | varchar | Store the product ID |
| **3** | Amount | varchar | Store the amount of a product user bought |

## 5. Interface design

**Table 14 Interface Design**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Purpose** | **Explain** |
| **1** | Login page | Login | A page for user to login with username and password. |
| **2** | Sign up page | Sign up | A page for new user to register with name, username, password, email. |
| **3** | Verify code | Verify code send to user email | User input the code which has been send to their email to verify account. |
| **4** | Home page | introduction | Contain shop information, introduction |
| **5** | Navigation bar | A navigation bar | Always show as a button to trigger at every page.  Allow user to choose different action such as:   * About: see the shop information * Admin: move to admin to manage products * Buy history: move to user’s buy history page * Shop now: move to shopping page * Logout: Sign out |
| **6** | Footer | Show information | Contain contact and shop information.  And always display at the bottom of every page. |
| **7** | About page | Detail information of website | Contain detail information of the website |
| **8** | Admin page | Admin page | A page for admin to manage list of products available.  Admin can add or delete  Edit button will direct to edit page which contain all information of selected product. |
| **9** | Edit page | Allow admin to edit product. | Contain all information of selected product.  Allow admin to adjust the information (except the ID) |
| **10** | Buy history page | List all products user has bought | List all products user has bought so that customers can review the bill or control their buying. |
| **11** | Shopping page | Shopping page | * Search bar: allow customers to search product by name * Brands filter: allow customers to filter products by brands. * Latest products: show 3 last product of the store. * Buy button: add the selected product to cart. * Click at the image of product: direct to the detail page of selected product. |
| **12** | Product detail page: | Show the detail of selected product. | * Show different images of product * Show name and price * Buy items button: add product to cart * Description: Some other feature of product. |
| **13** | Checkout page | Show cart and check out | * Show the list of product user has selected * Allow user to increase/decrease the quantity or remove entire product. * Show the total money * Allow customers to enter information: name, address for delivery, phone number. * Delivery button: generate invoice and store to database, send a confirm bill to customer address. |
| **14** | Confirm bill mail: | An email layout sends to customer to confirm the bill. | * An email layout sends to customer to confirm the bill. * Includes: Customer name, phone number, address, date, total money for customer services. |

# IV. Deploying and Testing:

## Scenario 1: Signup/Log in before conduct buying.

**User do not have account => Sign up => Verify code => Shopping.**

**Table 15 Scenario 1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Stage** | **Purpose** | **Explain** |
| **1** | Sign up  Input: customer information  Output: direct to verify page | Register | Get User information and send verification code to confirm account. |
| **2** | Verify code  Input: user get the code at email  Output:  Redirect to verify page if the code is wrong.  Move to homepage if the code is valid | Verify | Check the code user input. |
| **3** | Homepage: | Logged in | As can be seen at the navigation bar.  User has successfully sign up and login with email show as: hello + “user email” |

## Scenario 2: Conduct shoping

**User choose product => add to cart => check out**

**Table 16 Scenario 2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Stage** | **Purpose** | **Explain** |
| **1** | Select product and add to cart  Input: Product id  Output: direct to verify page | Shopping, add product to cart | If the buy button of product is clicked. That product will be added to cart with quantity = 1. |
| **2** | Checkout if the cart is empty | Check out | If the cart is empty then the checkout button will not appear. |
| **3** | Check out if the cart is not empty  Input: user name, address, phone number | Check out | User have to enter information for delivery and confirm bill. |
| **4** | Check the buy history | Review bill | User can check the list of buying product. |

## Scenario 3: Conduct shoping without signed in

**User choose product => add to cart => check out => sign in/sign up**

**Table 17 Scenario 3**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Stage** | **Purpose** | **Explain** |
| **1** | Check out page (when user not logged in) | Check out | If the user not logged in and click delivery button then the web will direct to sign in page |
| **2** | Sign in | Sign | Require user to sign in to conduct check out. |

## Scenario 4: Admin sign in to

**User choose product => add to cart => check out => sign in/sign up**

**Table 18 Scenario 4**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Stage** | **Purpose** | **Explain** |
| **1** | User login with admin account: | Login as admin | A admin button will appear on the navigation bar, direct to admin page |
| **2** | Admin page | Manage products | Admin can add, delete or edit product |

# V. Conclusion

## Pros:

Through the working progress, our group has learned and practice all of the subject which we have learn from the beginning to successfully build a shopping website.

The web after being deployed and tested is working properly, functionally. The speed response of the web is fast and productive.

All member has completed their assignment and do not have any conflict.

## Cons:

We used the template online.

The website is only had simple function and do not have much feature.

The database structure is uncomplicated.

## Future development:

We will re design the front-end of website to make it more beauty and add more function, feature so that the website can be use in business someday.

# Link source code:

[**https://github.com/NVA071002/IT\_Project.git**](https://github.com/NVA071002/IT_Project.git)

Our website has been deploy to Heroku:

**https://itprojectvanhthien.herokuapp.com**

# Reference Source:

Template of the website: <https://w3layouts.com>

Template of login/register : https://colorlib.com