**HO CHI MINH CITY UNIVERSITY TECHNOLOGY AND EDUCATION**

🞠◊🞠◊🞠



**INSTRUCTOR:**

**Nguyen Thien Bao**

**PROJECT ON SOFTWARE ENGINEERING**

**building a hi-tech online store**

**GROUP MEMBERS:**

**Ho Vu Minh Duc-17110022**

**Nguyen Quang Minh-17110051**

**Hoang Long Phan-17110046**

**SEMESTER: 1 – YEAR: 2020-2021**

**HO CHI MINH CITY**

# EVALUATION AND SCORE

## EVALUATION:

## SCORE:

|  |  |
| --- | --- |
| **WORD** | **NUMBER** |
|  |  |

# PREFACE

Our group are sincerely thankful to Mr. Nguyen Thien Bao – Lecturer of Faculty of Information Technology of Ho Chi Minh City University of Education and Technology. We often have some problems during implementation and testing. Thank to Mr. Bao, we could come over it and complete our project. We are grateful to him.

Thank you,

Mr.Bao

Nguyen Thien Bao

Table of Contents

[EVALUATION AND SCORE 2](#_Toc76665896)

[EVALUATION: 2](#_Toc76665897)

[SCORE: 2](#_Toc76665898)

[PREFACE 3](#_Toc76665899)

[I. Purpose 6](#_Toc76665900)

[II. Functionals Analysis 6](#_Toc76665901)

[1. Functional Business 6](#_Toc76665902)

[2. Functional System Requirements 7](#_Toc76665903)

[3. Non-functional system requirement 8](#_Toc76665904)

[III. Database system 9](#_Toc76665905)

[1. Database description 9](#_Toc76665906)

[ Users 9](#_Toc76665907)

[ Orders 9](#_Toc76665908)

[ Products 10](#_Toc76665909)

[ Reviews 11](#_Toc76665910)

[2. Database relationship 12](#_Toc76665911)

[IV. UI Design Of Hi-Tech Online Store 13](#_Toc76665912)

[1. GUI Flow 13](#_Toc76665913)

[2. Login Flow: 14](#_Toc76665914)

[3. Purchase Flow: 14](#_Toc76665915)

[4. User privileges flow: 15](#_Toc76665916)

[5. Screen design 16](#_Toc76665917)

[ Login screen 16](#_Toc76665918)

[ Register screen 16](#_Toc76665919)

[ Homepage 17](#_Toc76665920)

[ Product detail screen 17](#_Toc76665921)

[ Shipping address screen 18](#_Toc76665922)

[ Payment method screen 18](#_Toc76665923)

[ Place order screen 19](#_Toc76665924)

[ Order detail screen 19](#_Toc76665925)

[ User profile screen 20](#_Toc76665926)

[ User list screen 20](#_Toc76665927)

[ Product list screen 21](#_Toc76665928)

[ Order list screen 21](#_Toc76665929)

[V. Software design 22](#_Toc76665930)

[1. Identify actors and functions for use case diagrams 22](#_Toc76665931)

[2. Description of actors and functions 22](#_Toc76665932)

[2.1. Actors and its roles 22](#_Toc76665933)

[2.2. Functions and its roles 23](#_Toc76665934)

[3. System use case diagrams 25](#_Toc76665935)

[3.1. Use case descriptions 25](#_Toc76665936)

[Use Case - Login 25](#_Toc76665937)

[Use Case – Add new product 26](#_Toc76665938)

[Use Case – Edit product 26](#_Toc76665939)

[Use Case – Delete product 27](#_Toc76665940)

[Use Case – Change user information 27](#_Toc76665941)

[Use Case – Delete user 28](#_Toc76665942)

[Use Case – Change delivering status of an order 28](#_Toc76665943)

[Use Case – Make an order and purchase 29](#_Toc76665944)

[Use Case – Search product 29](#_Toc76665945)

[Use Case – View order history 30](#_Toc76665946)

[Use Case – Review order 30](#_Toc76665947)

[VI. Conclusions 31](#_Toc76665948)

[1. About knowledge and skill 31](#_Toc76665949)

[2. About topics 31](#_Toc76665950)

[3. Developing 31](#_Toc76665951)

[VII. Resources and References 32](#_Toc76665952)

[VIII. Work distribution table 33](#_Toc76665953)

1. Purpose**:**

During COVID-19 pandemic, many people do not go to store many more but buying stuff online on the internet, so the need of an website of online store is becoming more popular in these day

For that, we implement a website that selling hi-tech/electronics using MERN stack for satisfying people’s shopping need

1. Functionals Analysis**:**

## Functional Business

|  |  |  |
| --- | --- | --- |
| **Number** | **Function** | **Description** |
| 1 | Storage | Web’s database stores information about:  Order’s information: total price, product list, payment method, tax price, shipping price, …  User’s information: email, name, order history  Product’s information: name, description, price, quantity in stock, review, … |
| 2 | Searching | Search for the product’s information: name, description, price, quantity in stock, review, … |
| 3 | Authentication | System shall authenticate user and distinguish user role is an admin or not |
| 4 | Calculating | Total price of an order Shipping, tax fee User’s order history |
| 5 | Printing out | List of available products List of user’s orders, all orders List of active users List of active products |

## Functional System Requirements

|  |  |  |
| --- | --- | --- |
| **Number** | **Function** | **Description** |
| 1 | User Accounts | The customer enters the system as a user or guest. Check for available products, product’s information such as price,…  For customer who want to purchase product, they must login/register new account  Customer can edit his/her profile anytime |
| 2 | Registration of user profile | The system shall require a user to register, through 2 method, input information for register or register through social network through google or facebook, to carry out any transactions with it except for checking the list of products and its information. It will ask the user for the following information at the least – email, password |
| 3 | Checking product detail and availability | After logging in a user (either a registered user or a guest), the system shall show all information about the product user want to check – name, price, description, is out of stock or not, … |
| 4 | Making purchase | If user want to buy the product and not log on yet, he/she should login into system  In case the user wants to purchase, the system charges the price of the ticket to user’s credit card or e-wallet such as paypal. After payment, the system will mark the order payment successful status/fail, and admin will mark order delivered status is delivered or not if product is delivered |
| 5 | Checking order history | The system will list all of orders and its detail of user for user who want to checking order information |
| 6 | View order status | The system shall allow a user to view all information about his orders. After logging on, user can view order detail on purchase history. It retrieves the details of the order and presents them to the user in a convenient format, including any last minute changes, such as delivering and payment status |

## Non-functional system requirement

|  |  |  |
| --- | --- | --- |
| **Number** | **Function** | **Description** |
| 1 | Performance | Response time of the software should be less than 2 second most of the time. Response time refers to the waiting time while the software accesses, queries and retrieves the information from the databases |
| 2 | Reliability | System shall be available 24 hours a day, 7 days a week System shall always provide real time information about order information.  System shall validate input and authenticate user if is valid or not |
| 3 | Integrity | Users need to be authenticated before having access to any personal data  System shall authenticate if user is admin or not to provide access to designated features |
| 4 | Usability | System shall provide a easy-to-use graphical interface similar to other existing ecommerce system so that the users do not have difficulty learning using it  Any notification or error messages generated by system shall be clear, polite |

# Database system

## Database description

### Users

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Value** | **Notes** |
| id | string | Primary Key | Automated created by DB System |
| name | string |  | Required |
| isAdmin | boolean |  | Default value: false |
| email | string |  | Required |
| password | string |  | Required |
| createdAt | date |  | Automated created by DB System |
| updatedAt | date |  | Automated created by DB System |

### Orders

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Value** | **Notes** |
| id | string | Primary Key | Automated created by DB System |
| userId | string | Foreign Key |  |
| taxPrice | number |  |  |
| totalPrice | number |  |  |
| shippingPrice | number |  |  |
| isPaid | boolean |  | Default value: false |
| isDelivered | boolean |  | Default value: false |
| orderItems | Products[] |  |  |
| shippingAddress | string{} |  |  |
| paymentMethod | string |  |  |
| paymentResult | string{} |  |  |
| deliveredAt | date |  |  |
| createdAt | date |  | Automated created by DB System |
| updatedAt | date |  | Automated created by DB System |

### Products

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Value** | **Notes** |
| id | string | Primary Key | Automated created by DB System |
| userId | string | Foreign Key |  |
| rating | number |  | Default value: 0 |
| numReviews | number |  | Default value: 0 |
| price | number |  |  |
| countInStock | number |  |  |
| name | string |  |  |
| image | string |  |  |
| brand | string |  |  |
| category | string |  |  |
| description | string |  |  |
| reviews | Reviews[] |  |  |
| createdAt | date |  | Automated created by DB System |
| updatedAt | date |  | Automated created by DB System |

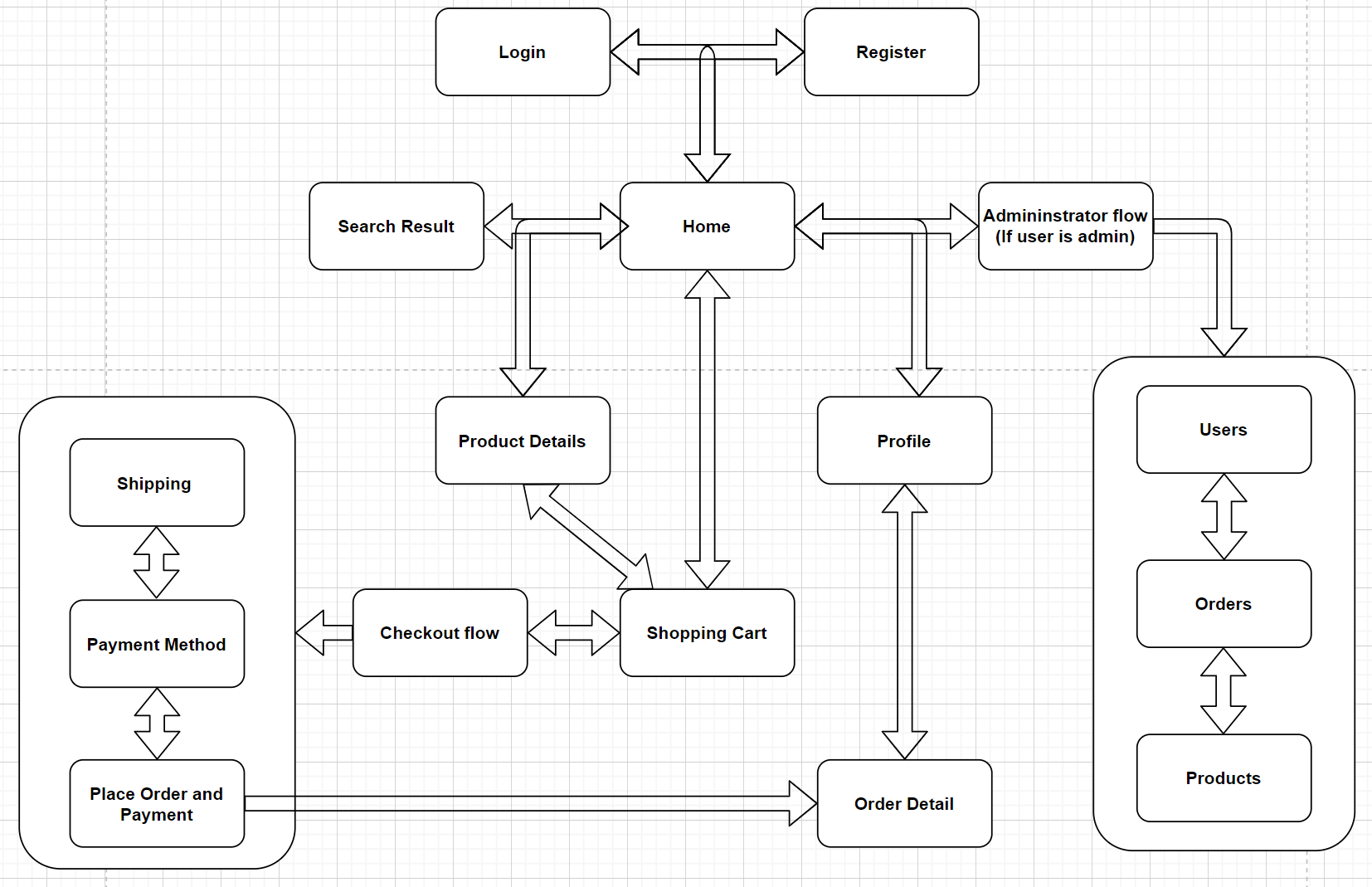
### Reviews

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Value** | **Notes** |
| id | string | Primary Key | Automated created by DB System |
| name | string |  |  |
| rating | string |  |  |
| comment | string |  |  |
| userId | string | Foreign Key |  |
| createdAt | date |  | Automated created by DB System |
| updatedAt | date |  | Automated created by DB System |

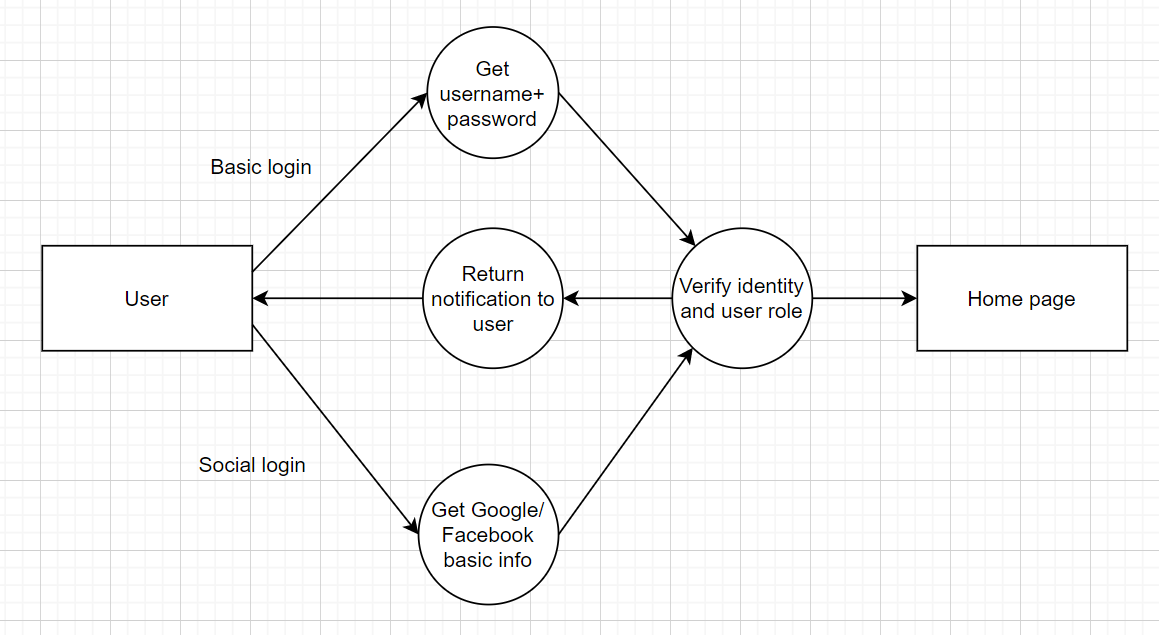
## Database relationship

# UI Design Of Hi-Tech Online Store

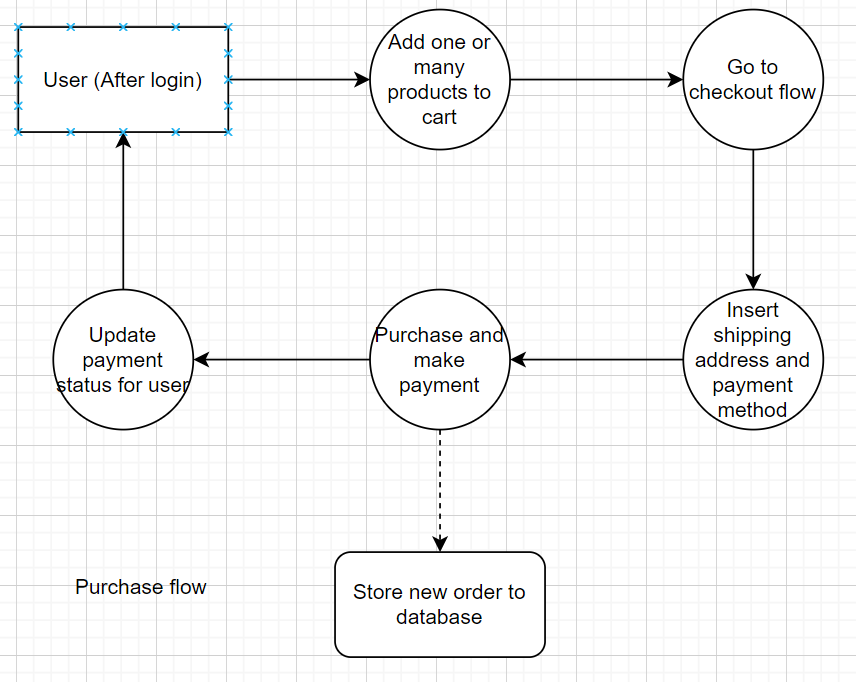
## GUI Flow



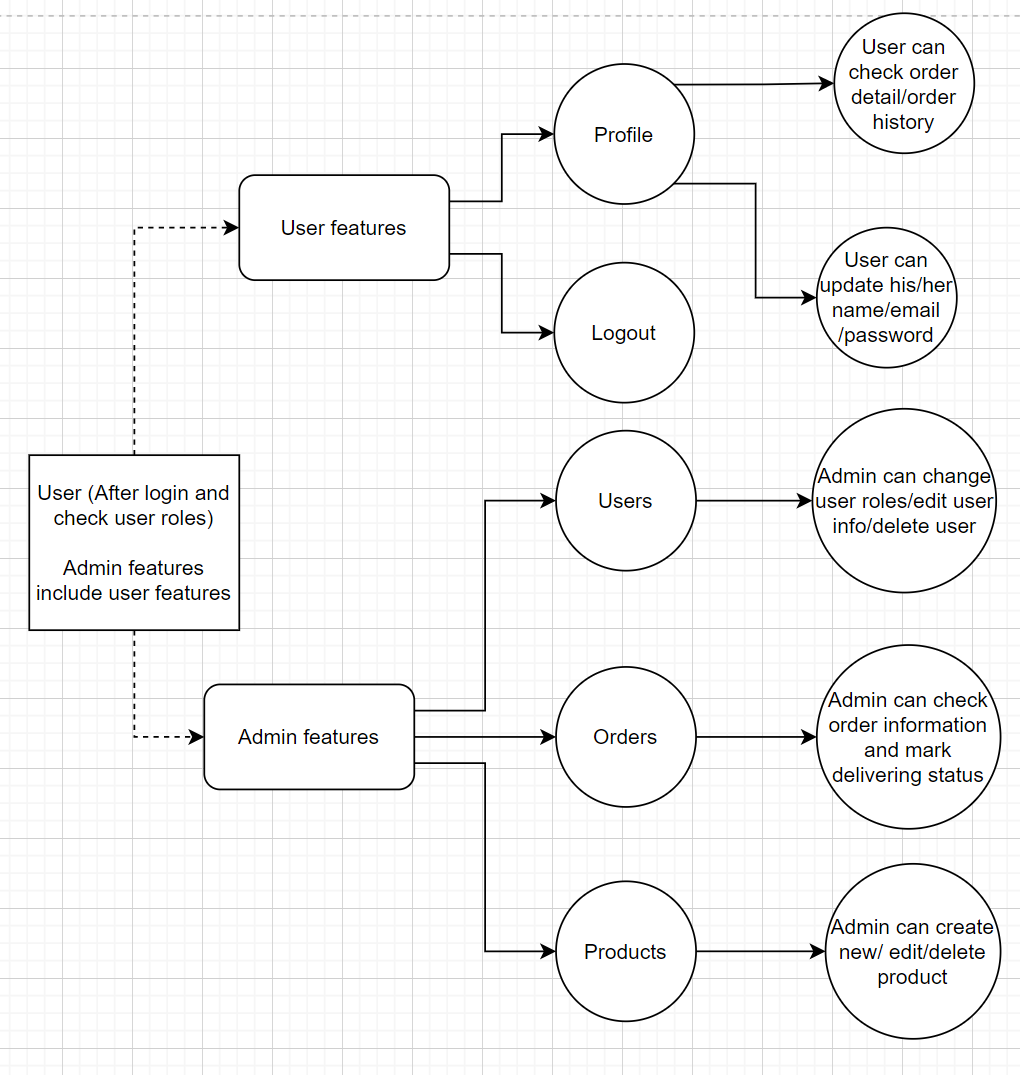
## Login Flow:



## Purchase Flow:

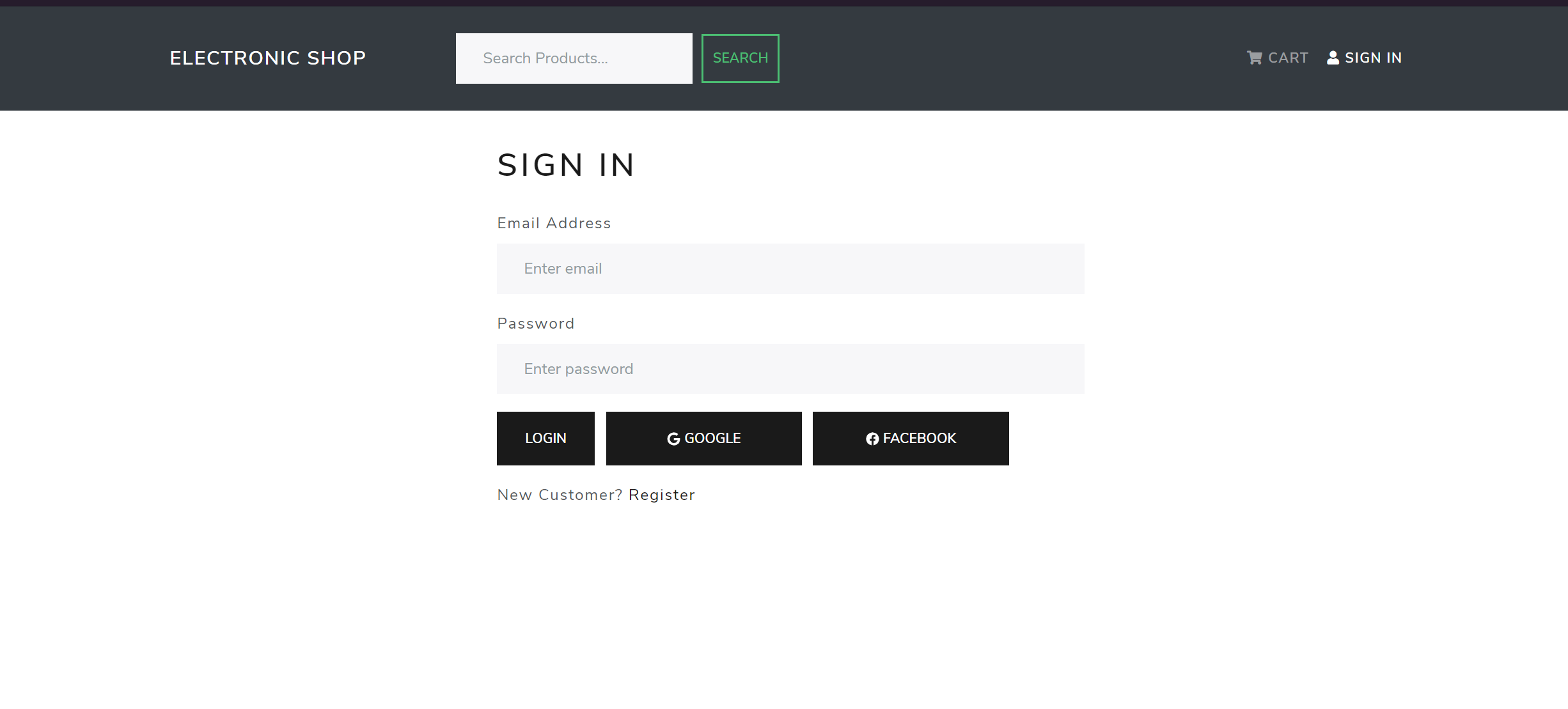


## User privileges flow:

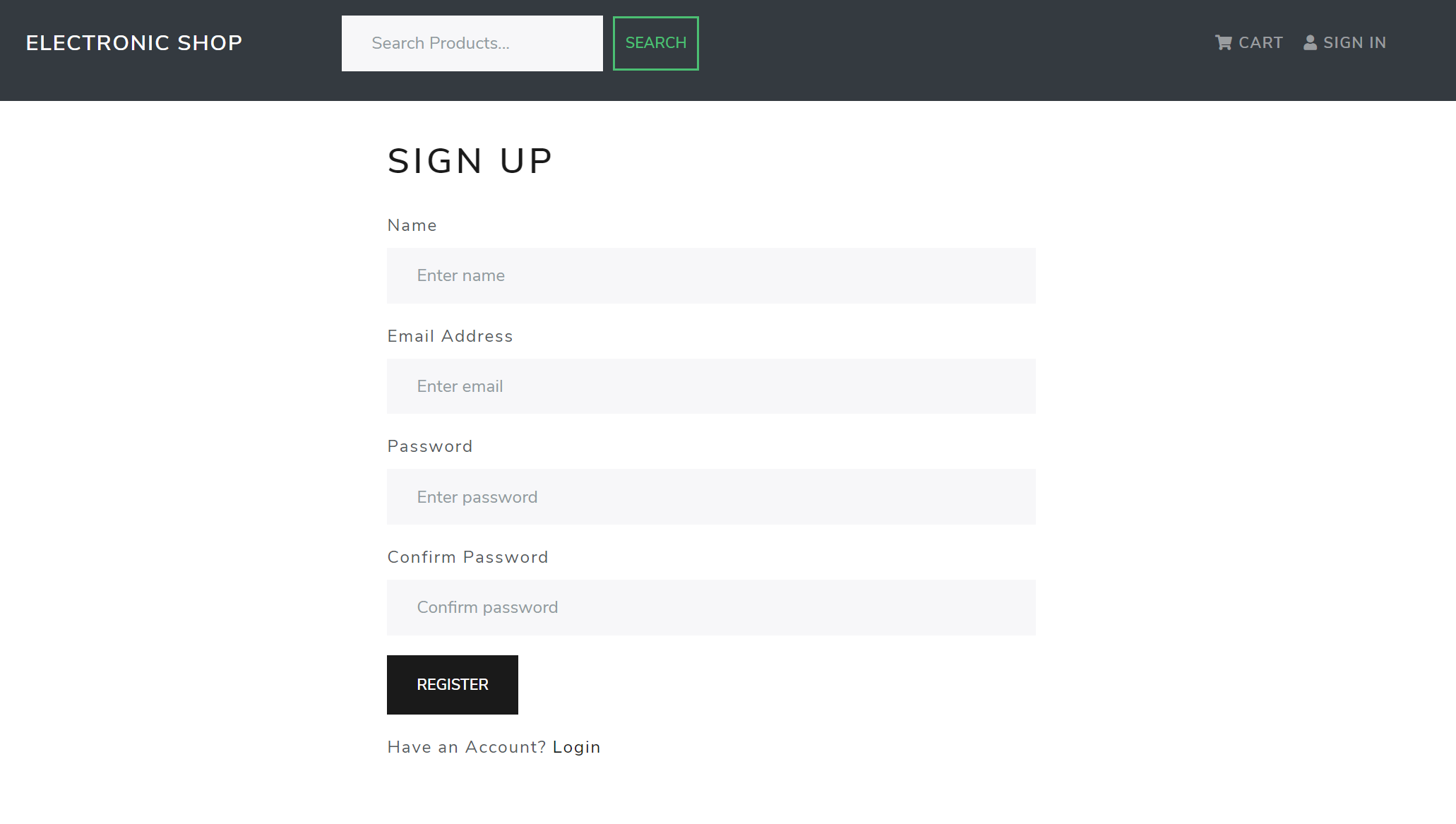


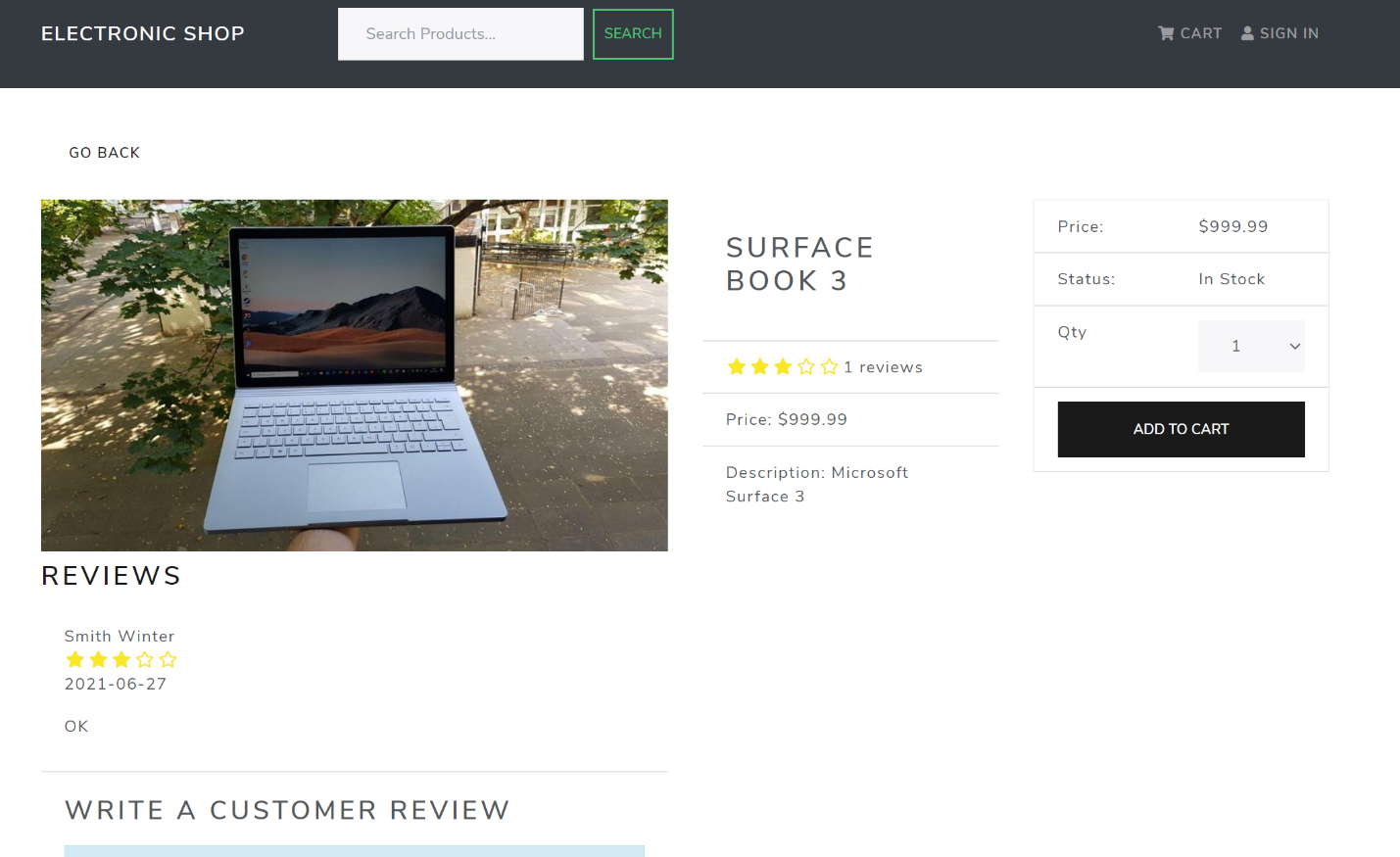
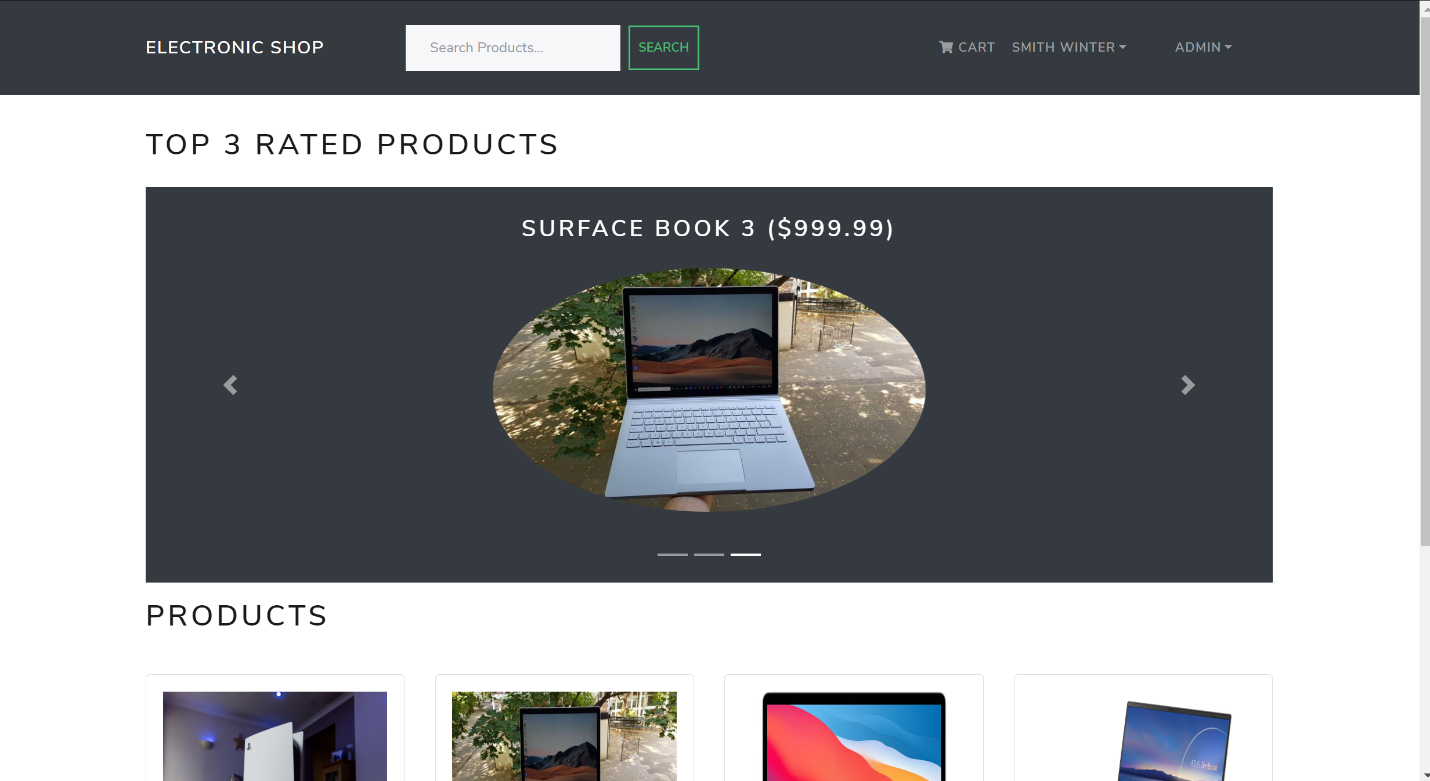
## Screen design

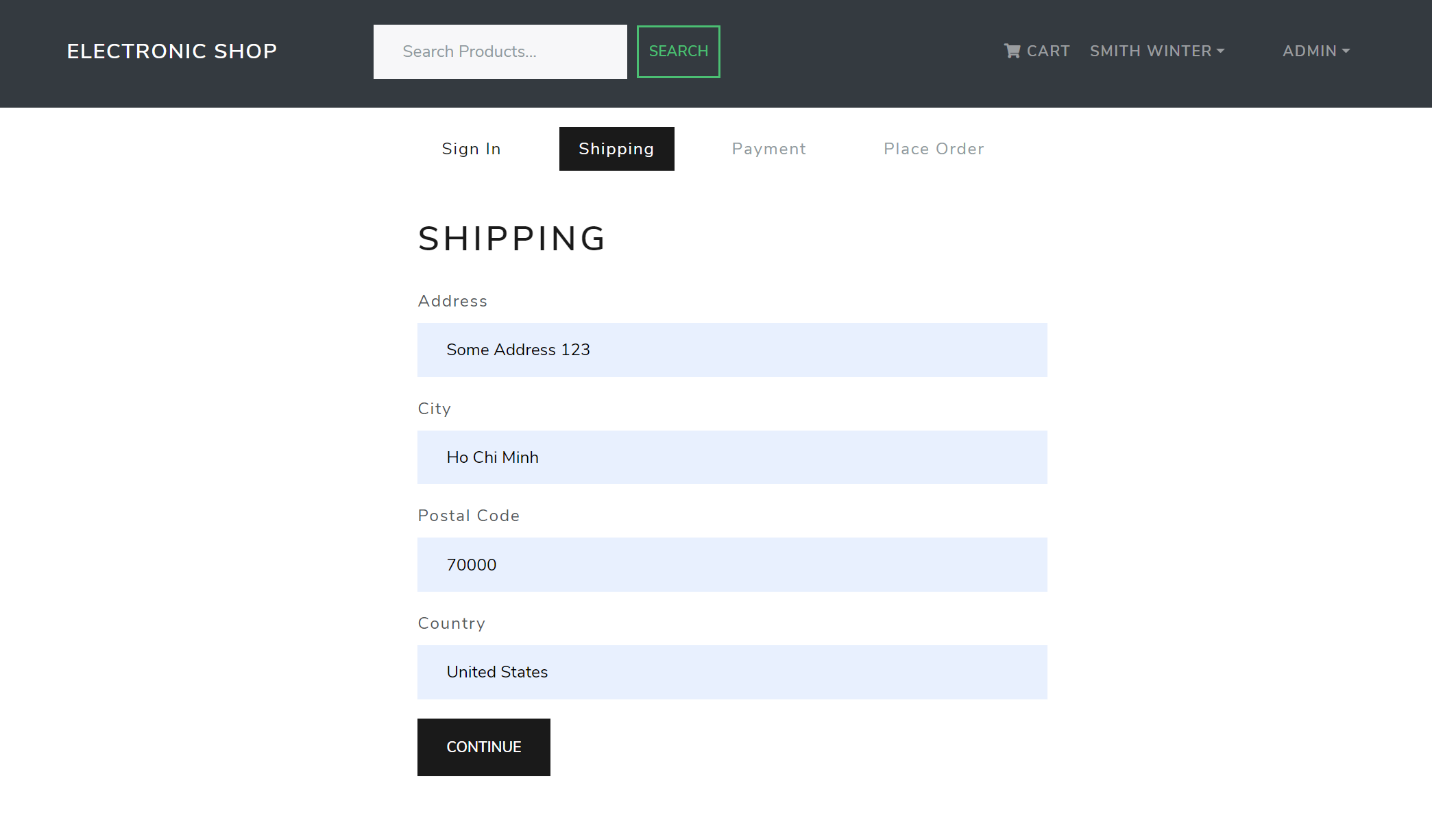
* Login screen**:** In this screen user input username and password to login, another way to login is through social network (google, facebook)



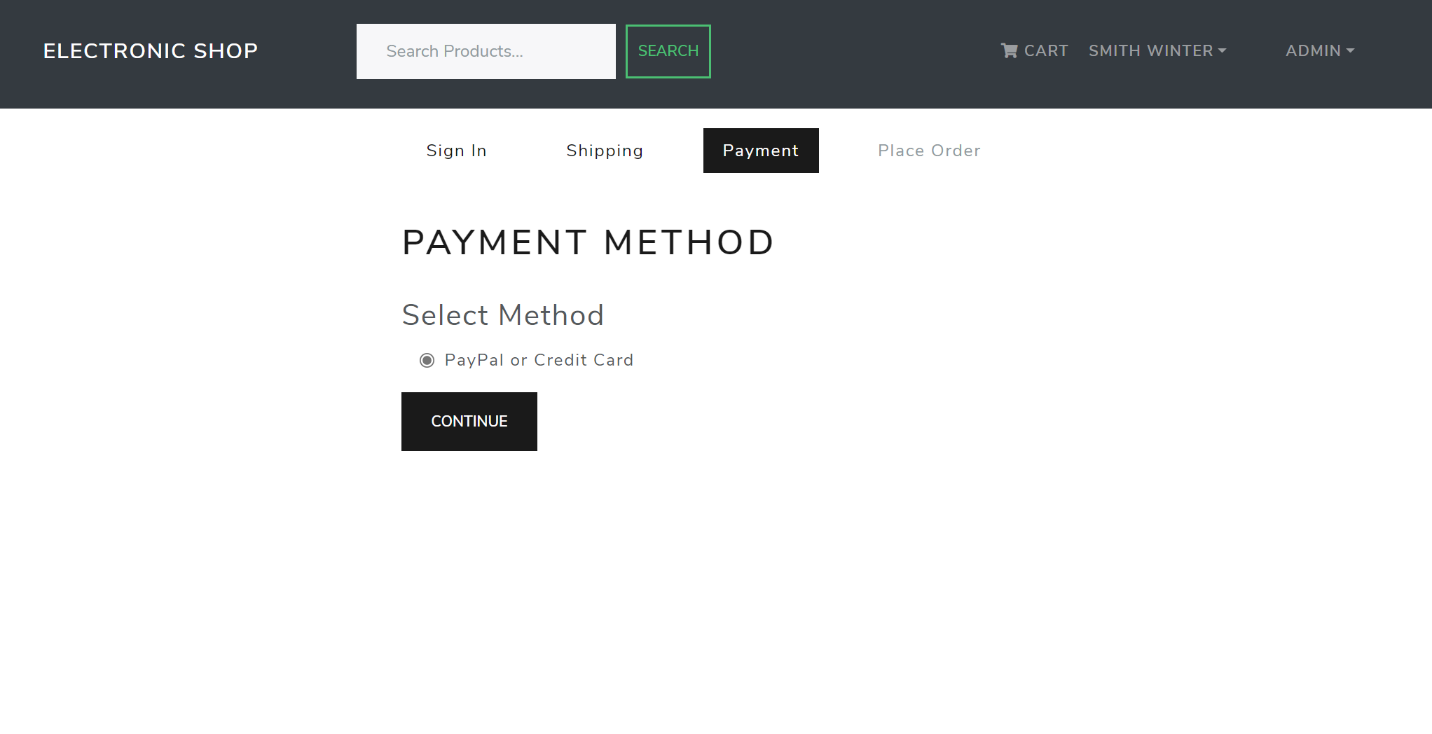
* Register screen**:** In this screen user input username and password to register

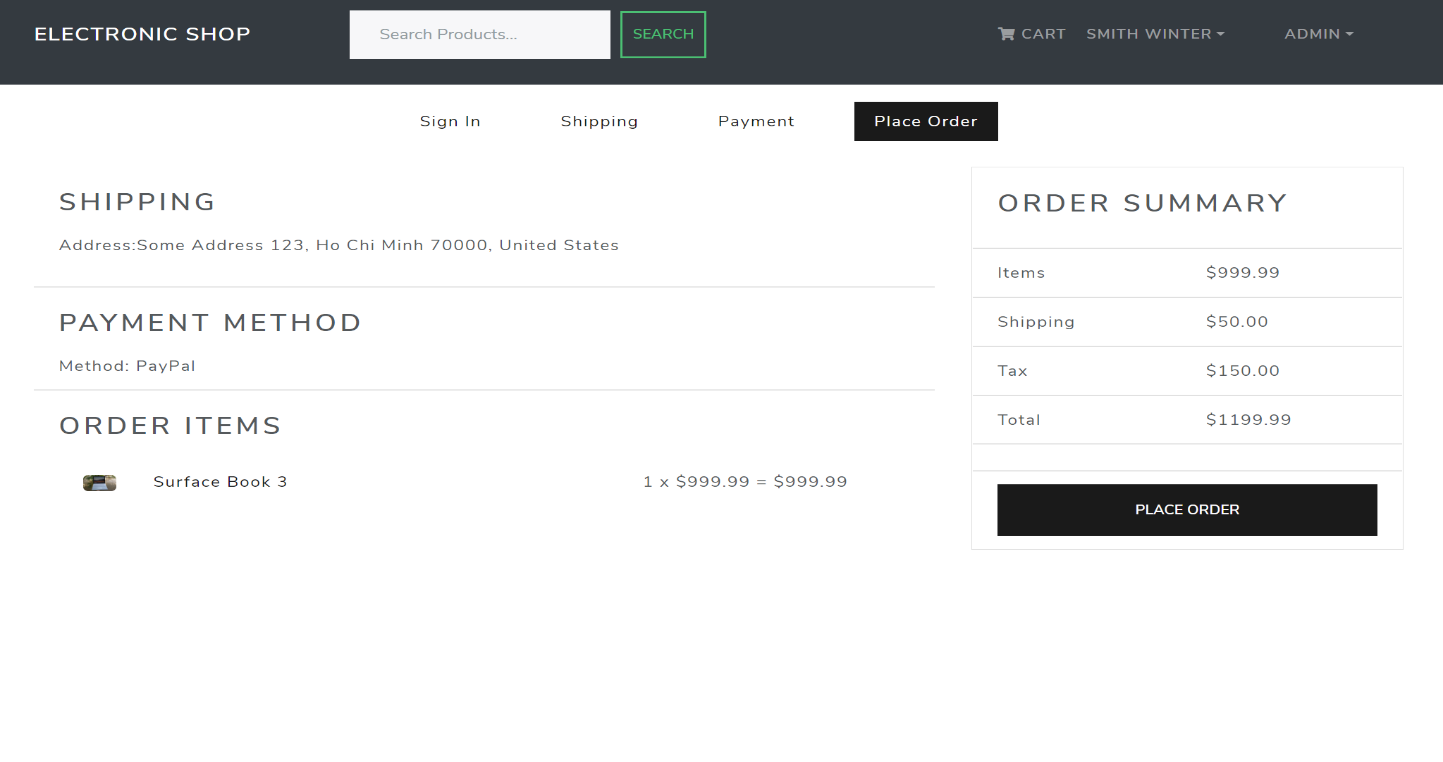
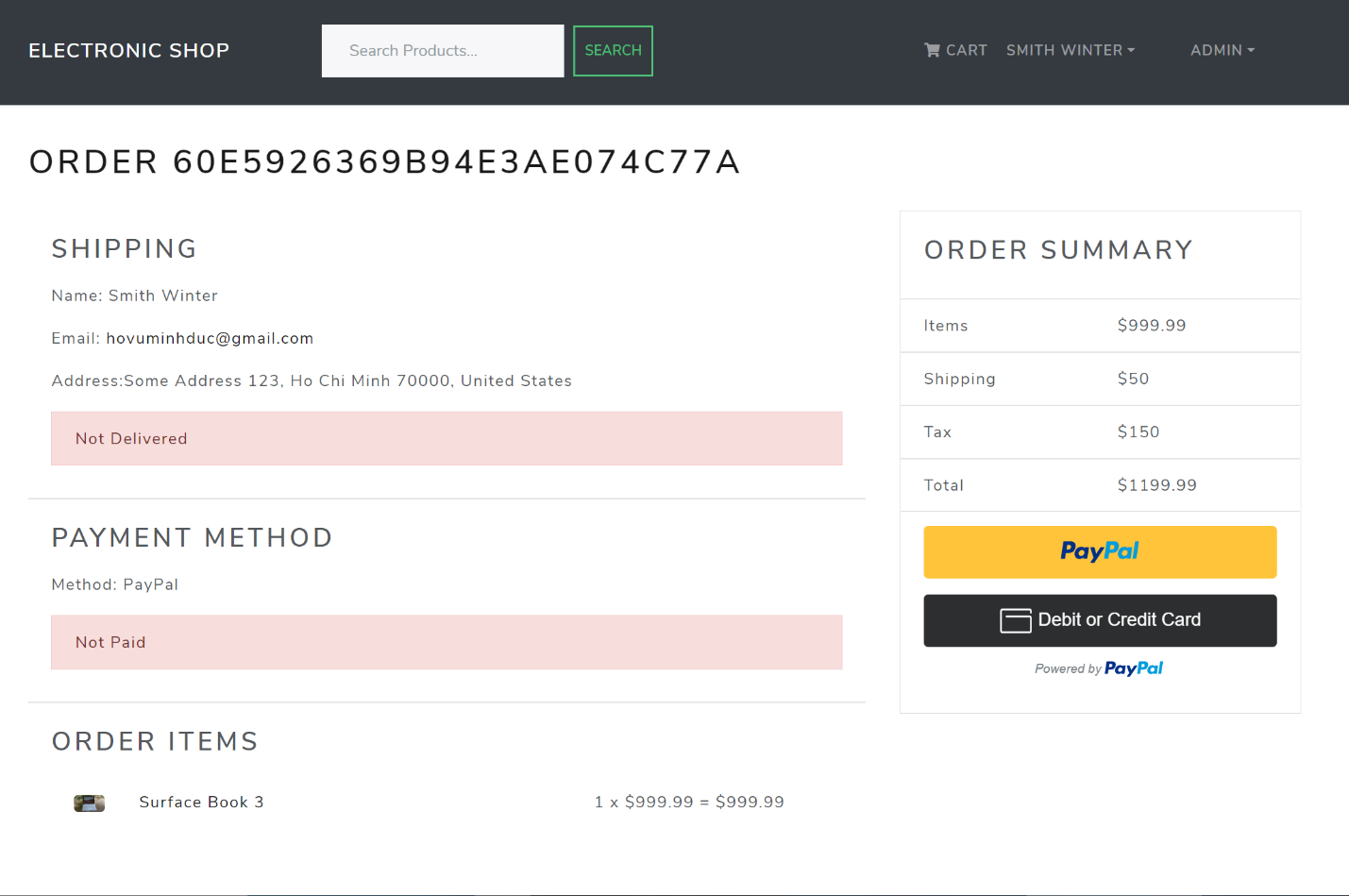
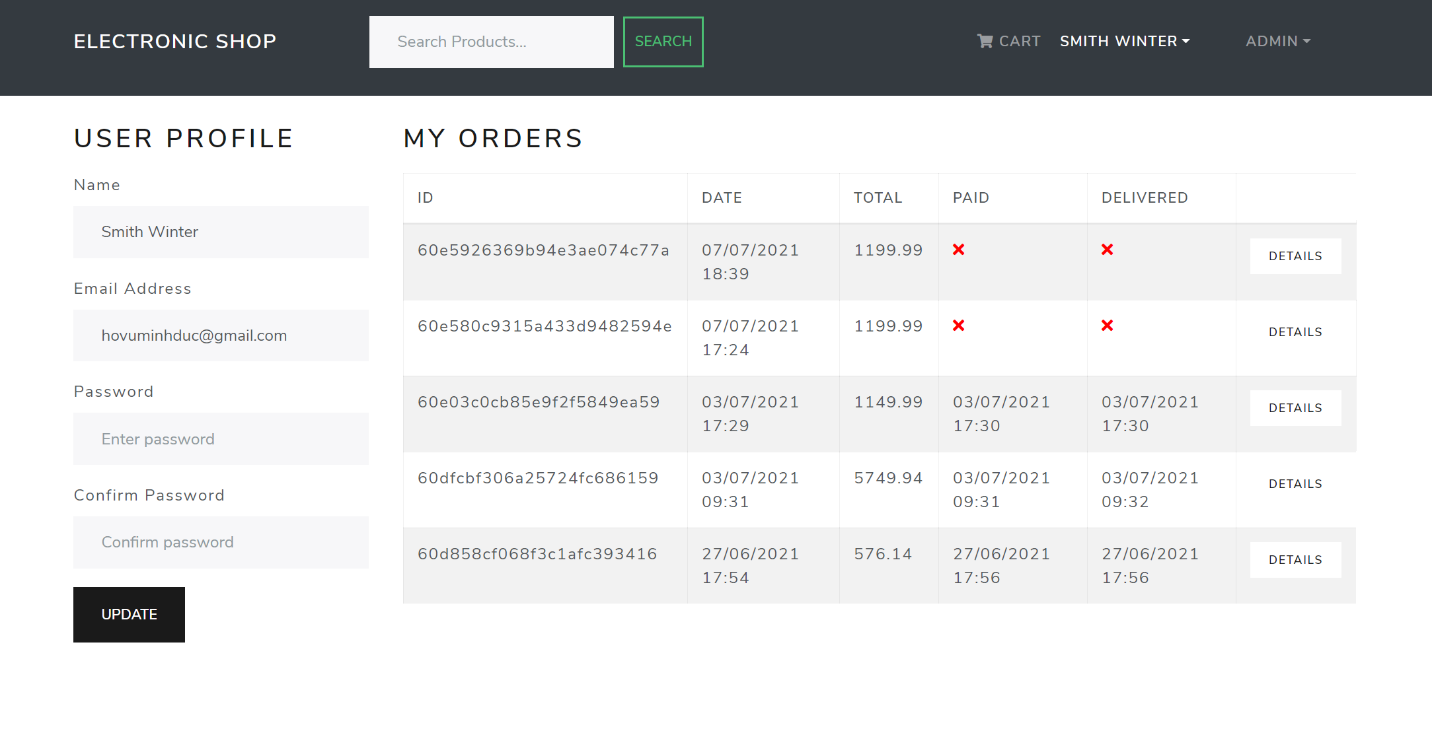


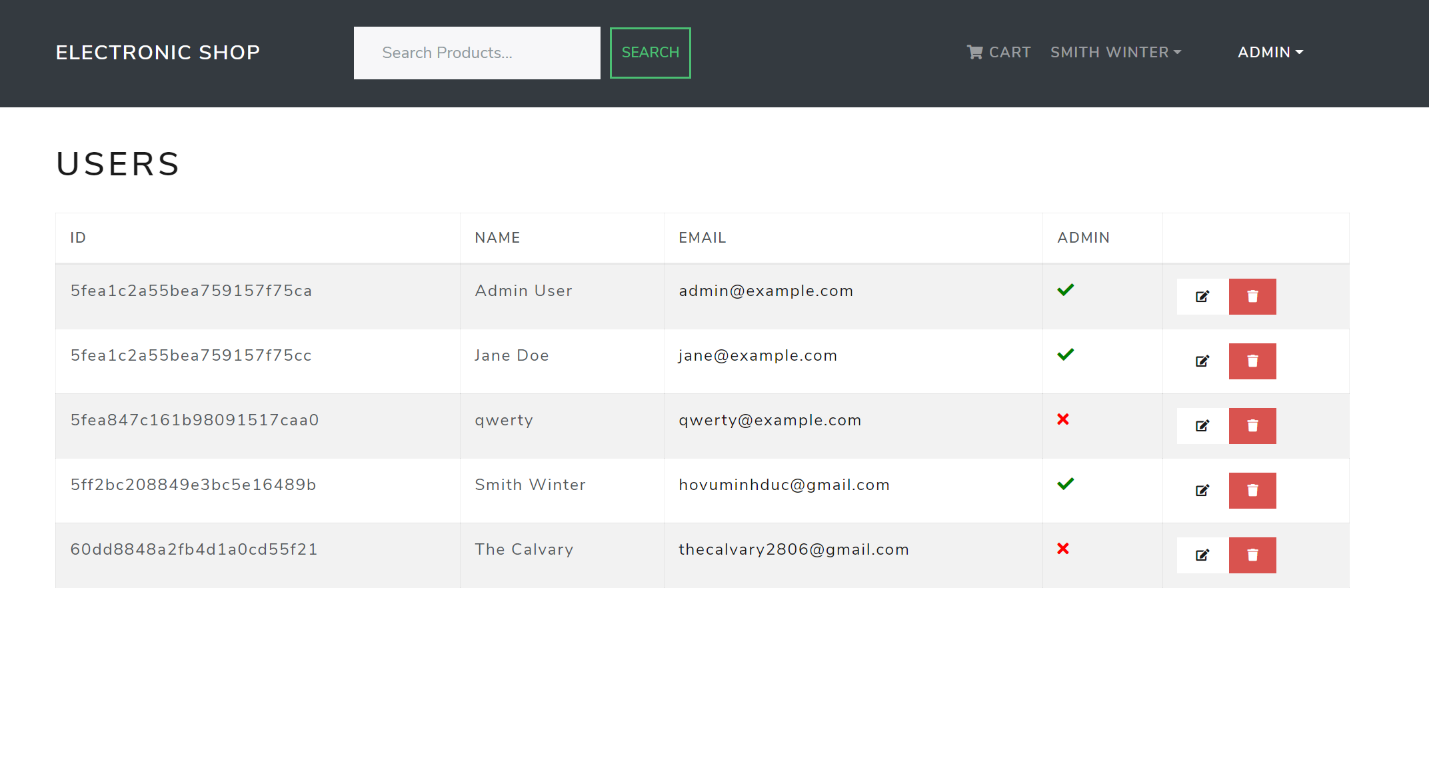
* Homepage**:** In this screen, there are carousel display top rated products and products list
* Product detail screen**:** In this screen display product information and user review
* Shipping address screen**:**  In this screen, user fills information to proceed to payment method screen



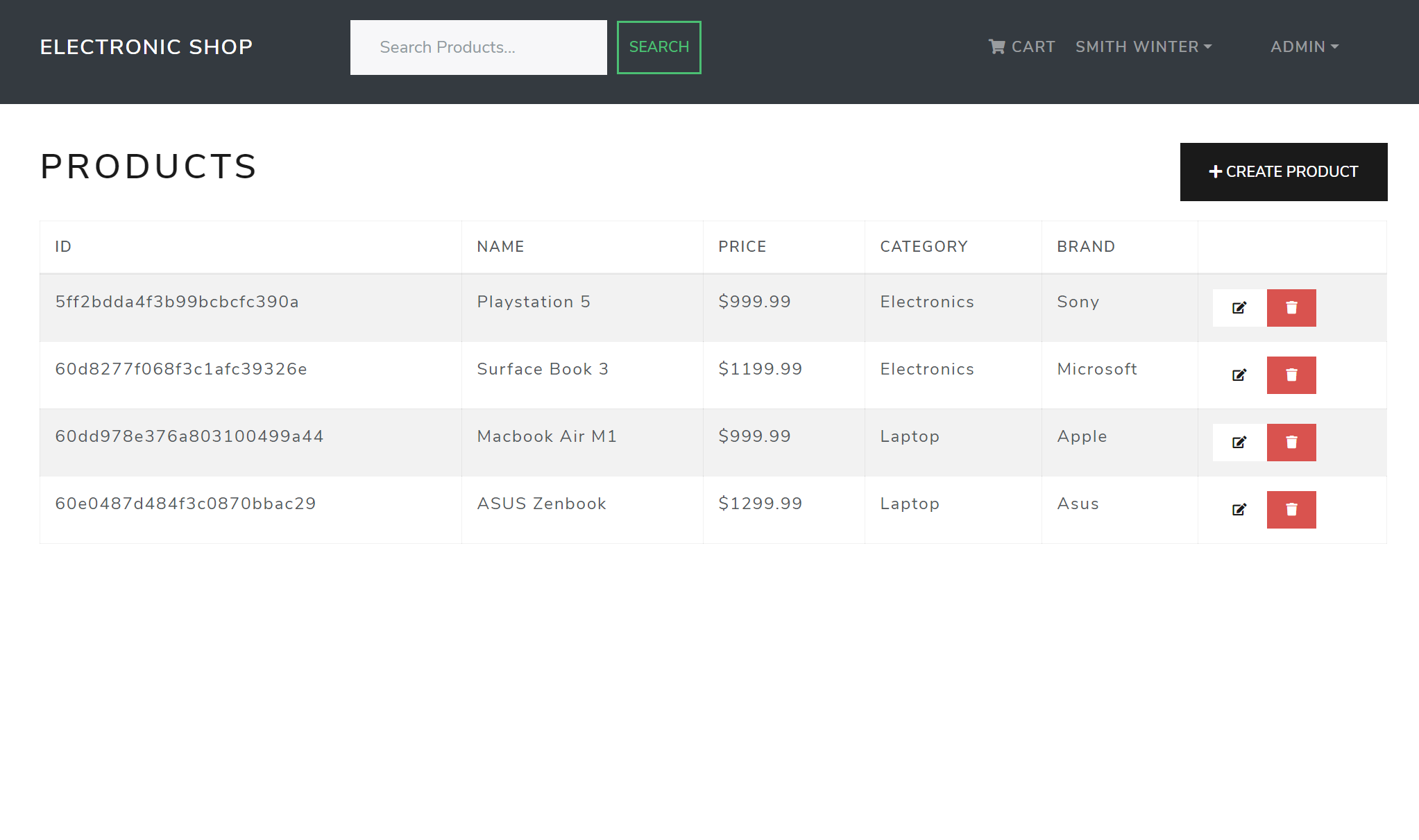
* Payment method screen**:** In this screen, user select payment method



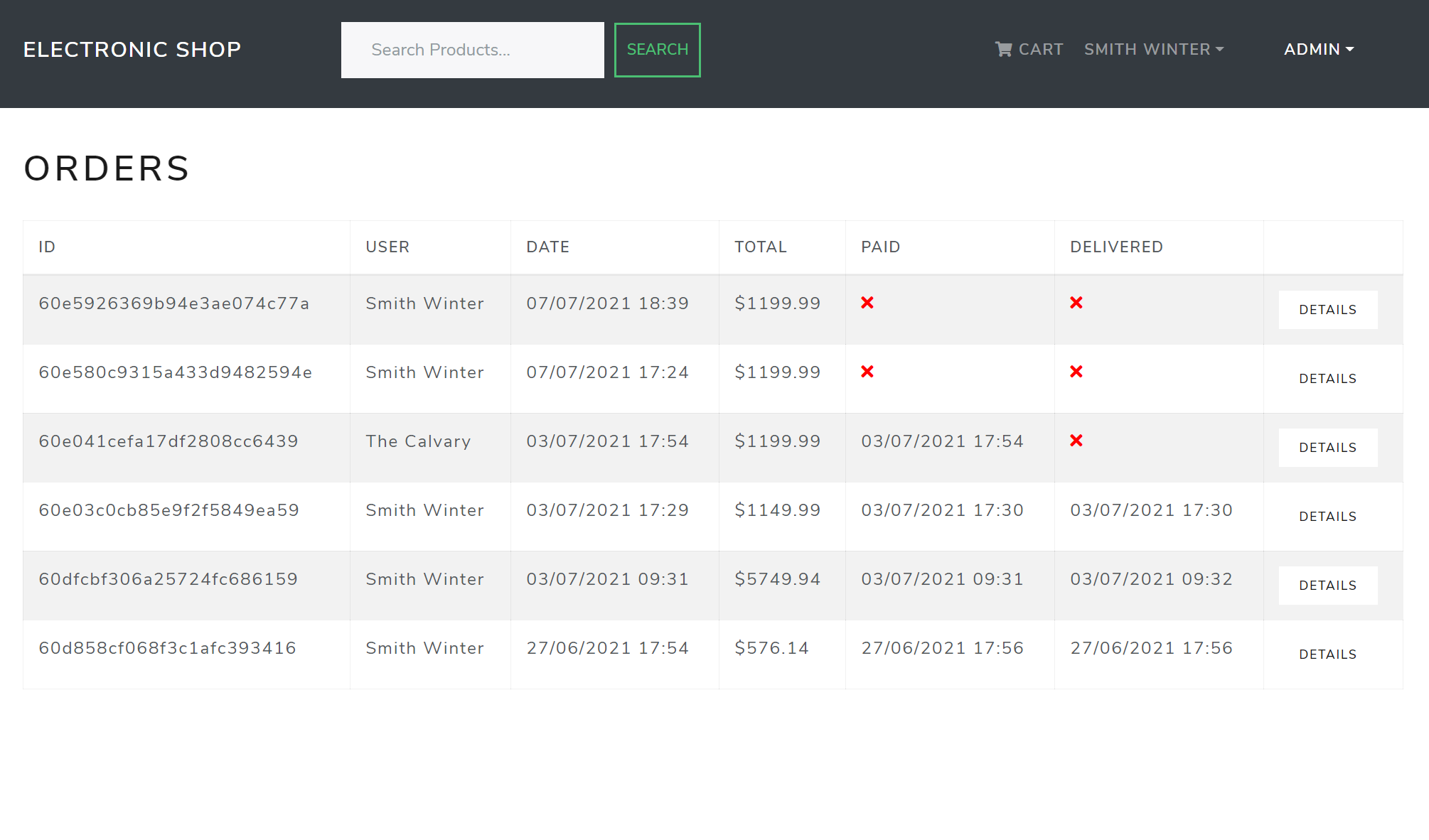
* Place order screen**:** In this screen, user check information of an order then place order
* Order detail screen**:** This screen display information of an order
* User profile screen**:** In this screen, user can edit information and view order history
* User list screen**:** In this screen, admin can view/edit user’s information or delete user



* Product list screen**:** In this screen, admin can view/edit product’s information or delete product



* Order list screen**:** In this screen, admin can view all order



# Software design

## Identify actors and functions for use case diagrams

|  |  |
| --- | --- |
| **Actors** | **Functions** |
| Admin | * Login * Product management * User management * Order management * Statistics |
| Customer | * Login * Profile management * Search product * View product information * Review product * Make an order * Make a payment for an order * Statistics |

## Description of actors and functions

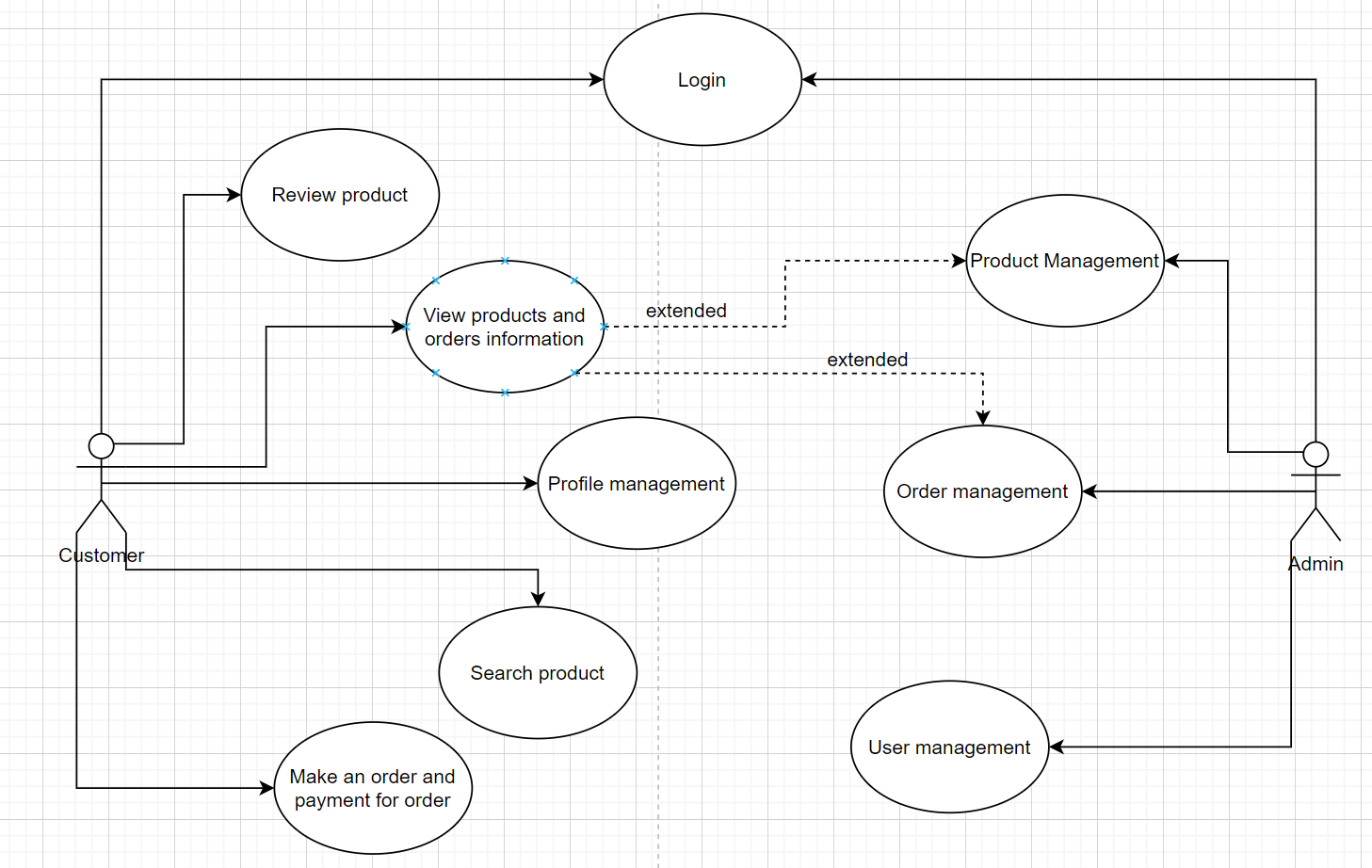
### Actors and its roles

|  |  |
| --- | --- |
| **Actors** | **Roles** |
| Admin | * Login system * Add new product * Edit product * Delete product * Change user information * Delete users * View product list * View order list * View user list * Change delivering status of an order |
| Customer | * Login system * Search product * View product information * Make an order and purchase * View order history * Change user information * Review product |

### Functions and its roles

|  |  |  |
| --- | --- | --- |
| **ID** | **Functions** | **Roles** |
|  | Login system | Authenticate user or admin login to system |
|  | Add new product | Admin input new product information to the system, such as name, description, price, image, … in admin panel |
|  | Edit product | Admin update product information to the system, such as name, description, price, image, … in admin panel |
|  | Delete product | Admin deactivate the product from the system in admin panel |
|  | Change user information | Customer can change information such as name, email, password on his own account  Admin can change information such as name, email password excludes user roles on his own  Admin can change information such as name, email password, roles on other account |
|  | Delete users | Admin can deactivate another account (not include his own account) in admin panel in case of special case such as agreement violation |
|  | View product list | Admin can view all products in admin panel |
|  | View order list | Admin can view all orders in admin panel |
|  | View user list | Admin can view all users in admin panel |
|  | Change delivering status of an order | Admin can go to order detail and change delivering status of an order from not delivered to delivered |
|  | Search product | User can search product available in system, after searching, display list of product and its detail found base on keyword |
|  | View product information | User can check product information such as checking if the product is out of stock or not |
|  | Make an order and purchase | After making an order, user can make payment by using Paypal payment gate |
|  | View order history | User can view list of order history and its detail |
|  | Review product | User can review a product and provide feedback to it |

## System use case diagrams



### Use case descriptions

|  |  |
| --- | --- |
| Use Case - Login | |
| **Description** | Allow admin and customer login to the system |
| **Actors** | Customer, admin |
| **Pre-conditions** | Customer and admin both know email, password (optional) for login through form or through social network |
| **Steps** | 1. Admin/Customer go to website 2. Input email and password 3. Press “Login” Button and enter / Press “Google” or “Facebook” to login 4. If login success and user account is not deactivated, direct the user to home page, if not, return notifications to user |
| Use Case – Add new product | |
| **Description** | Allow admin creates new product in system |
| **Actors** | Admin |
| **Pre-conditions** | Admin must be logged in to use this function |
| **Steps** | 1. Admin go to website and go to admin’s product management page 2. Input information to form, such as name, description, price, … 3. After input, press “Add” button 4. If product create successful, new product will be updated to database and display new product in list, otherwise, return error and input again |

|  |  |
| --- | --- |
| Use Case – Edit product | |
| **Description** | Allow admin edit selected product in system |
| **Actors** | Admin |
| **Pre-conditions** | Admin must be logged in to use this function |
| **Steps** | 1. Admin go to website and go to admin’s product management page, press edit button to edit selected product 2. Input information to form, such as name, description, price, … 3. After input, press “Update” button 4. If product edit successful, selected product will be updated to database and display notification, otherwise, return error and input again |

|  |  |
| --- | --- |
| Use Case – Delete product | |
| **Description** | Allow admin delete selected product in system |
| **Actors** | Admin |
| **Pre-conditions** | Admin must be logged in to use this function |
| **Steps** | 1. Admin go to website and go to product management page, press delete button to delete selected product 2. Press “Yes” to confirmation box 3. Product has been deleted and system return notification to user |

|  |  |
| --- | --- |
| Use Case – Change user information | |
| **Description** | Allow admin and customer edit account information |
| **Actors** | Admin, Customer |
| **Pre-conditions** | Admin/Customer must be logged in to use this function |
| **Steps** | * **For customer**  1. Customer go to profile page 2. Customer change information such as name, email, password 3. Customer press “update” button to update the user’s info to database 4. Database will save edit info and return notifications to user  * **For admin**  1. Admin go to user management page 2. Admin change information such as name, email on all accounts but cannot change his own roles 3. Admin press “update” button to update the user’s info to database 4. Database will save edit info and return notifications to user |

|  |  |
| --- | --- |
| Use Case – Delete user | |
| **Description** | Allow admin delete user account (exclude his own) |
| **Actors** | Admin |
| **Pre-conditions** | Admin must be logged in to use this function |
| **Steps** | 1. Admin go to user management page 2. Admin press “Delete” button to selected account 3. Press “Yes” to confirmation box 4. System return user delete successful |

|  |  |
| --- | --- |
| Use Case – Change delivering status of an order | |
| **Description** | Allow admin change delivering status of an order in the system |
| **Actors** | Admin |
| **Pre-conditions** | Admin must be logged in to use this function |
| **Steps** | 1. Admin go to order management page 2. Press “Detail” to navigate to selected order detail 3. Admin press “Mark as delivered” button to change order’s delivering status |

|  |  |
| --- | --- |
| Use Case – Make an order and purchase | |
| **Description** | Allow customer to make an order and make payment for it |
| **Actors** | Customer |
| **Pre-conditions** | Customer must be logged in to use this function  Shopping carts must not be empty |
| **Steps** | 1. Customer fill shipping address form, such as name, address, zip code 2. Customer select payment method 3. Customer check order information for confirmation and make an order 4. Customer make payment for order by using Paypal payment gate |

|  |  |
| --- | --- |
| Use Case – Search product | |
| **Description** | Allow customer or guest search a product |
| **Actors** | Guest, Admin, Customer |
| **Pre-conditions** | None |
| **Steps** | 1. Admin/Guest/Customer input keyword to search box 2. Press “Search” button to search 3. System will display products that relevant to keyword |

|  |  |
| --- | --- |
| Use Case – View order history | |
| **Description** | Allow customer view his order history |
| **Actors** | Customer |
| **Pre-conditions** | Customer must be logged in to use this function |
| **Steps** | 1. Customer go to profile page 2. Customer view order history list and press “Details” button to view selected order detail |

|  |  |
| --- | --- |
| Use Case – Review order | |
| **Description** | Allow customer review an order |
| **Actors** | Customer |
| **Pre-conditions** | Customer must be logged in to use this function |
| **Steps** | 1. User select a product to navigate to product detail 2. User select number of stars and input comment 3. User press “Review” button to upload new review or overwrite previous review of a product to database |

# Conclusions

## About knowledge and skill

Through the time we design and implement hi-tech online store, we have learned many things about teamwork, researching, documentation

* Learn about ReactJS, frontend framework and its library such as React Bootstrap
* Learn about MongoDB database and how to implement back-end with database by using NodeJS with ExpressJS as framework
* Learn about Redux for implementing frontend state management

## About topics

We build the online store have basic functions of an ecommerce store

* With guest: Guest can visit the store without login
* With customer: Customer can make an order, payment for order, review product and manage profile
* With admin: Admin can manage orders, products and users

However, we have some limitations due to the time limit, such as searching functions for admin and user auto logout after edit profile

## Developing

This is first time we implement the whole app, including frontend and backend, so we have some mistake due to lack of experience. If we have more time, we will improve UI/UX of the store, implement some functions such as calculate revenue in weekly, monthly, yearly, export receipt to user, backup database schedule, …

# Resources and References

* ReactJS: [React – A JavaScript library for building user interfaces (reactjs.org)](https://reactjs.org/)
* React Bootstrap: [React-Bootstrap · React-Bootstrap Documentation](https://react-bootstrap.github.io/)
* Redux: [Redux - A predictable state container for JavaScript apps. | Redux](https://redux.js.org/)
* NodeJS: [Node.js (nodejs.org)](https://nodejs.org/en/)
* ExpressJS: [Express - Node.js web application framework (expressjs.com)](https://expressjs.com/)
* MongoDB: [The most popular database for modern apps | MongoDB](https://www.mongodb.com/)
* Paypal Developer Docs: [PayPal Developer](https://developer.paypal.com/home)
* **Source code**: [SmithWinter/TLCN: ElectronicShop (github.com)](https://github.com/SmithWinter/TLCN)

# Work distribution table

|  |  |  |
| --- | --- | --- |
| **Member name** | **Task Assignment** | **Report Assignment** |
| Ho Vu Minh Duc | 1. Create MongoDB database instances on its online service 2. Implement database model code 3. Implement backend middleware and API routes with functions 4. Implement Paypal payment gate | Demo and explain the program |
| Nguyen Quang Minh | 1. Implement screens of the website 2. Design GUI and UI of the website 3. Implement state management code using Redux | Write the report |
| Phan Hoang Long | 1. Design GUI and UI of the website 2. Implement state management code using Redux 3. Implement screens of the website | Write the report |
| All member | 1. Research ReactJS, Redux, MongoDB, React Bootstrap, NodeJS and ExpressJS 2. Testing through the website 3. Integrate backend to frontend |  |