Hienlth’s AGILE Course – Group 2

**Online Ordering System for the Daily Grind Café**

# Part 1: Project overview

## Group structure and Bitbucket structure

Team structure:

Group 2 was divided into 2 teams:

* + Front-end team:
    - * Nguyễn Thái Sang
      * Lê Quang Huy
      * Lê Thanh Tuấn
  + Back-end team:
    - Phan Huỳnh Văn Ngọc
    - Lê Trường Thanh
    - Nguyễn Hoàng Kỳ Châu
    - Phan Minh Tâm

Total Bitbucket accounts: 6

Bitbucket commit list:

* + - * Lê Quang Huy [unknown <lequanghuy>]
    - Phan Huỳnh Văn Ngọc [vanngoc <vanngoc>]
    - Lê Trường Thanh [thanhlth <thanhlth4301104162>]
    - Nguyễn Hoàng Kỳ Châu & Phan Minh Tâm [DESKTOP-C3NHQP]

## System Specification and Database Design (CSO-22, CSO-23)

## System Specification

* Programming Language: PHP
* Working environment: Xampp
* Framework used: Laravel 8.x
* Database: MySQL that is bundled in Xampp

### Source Code

* Version Control: BitBucket
* Link: <https://bitbucket.org/QuangHuyLe/coffee-ordering-system/src/master/>

### Predicted Webpages

No authentication required:

* Home page
  + Show recommended products
  + Show new products
  + Show promotions, if exists
  + Be mindful of each product’s in stock number
* Product List page
  + Show everything in either grid view or list view, be mindful of each product’s in stock number
* Product Detail page
  + Let the customer adds this item into their carts if it’s still in stock, with some customizable attributes:
    - Quantity(number of cups)
    - Cup size: Sm/M/L
    - Flavor: sweetness and milk
  + Show related products
* Cart page
  + Let the customer remove any or all items in their cart
  + Let the customer proceed to checkout when their cart exists
* Checkout page
  + If the customer is anonymous, requires them to provide their personal information (name, phone number, email), else uses the information they provided during their registration.
  + Let the customer choose their shipping method (pickup or deliver); if deliver is chosen, they are required to input delivery time (either default or custom) and delivery address
  + Optional: let the customer proceed to a checkout service (i.e. PayPal)
* Login/Logout page
  + Other than username and password, the customer is prompted to input their name, phone number, email and address
  + Optional: verification with phone number or email
  + Optional: password recovering

Authentication required:

* User’s Order List page
  + User can view all of their orders, each contains the products, prices and order’s status (pending/processing/shipping/delivered/cancelled)
  + User can notify the store if an order is not delivered.
    - Optional: notify the shop via email.
* Administrator’s Order List page
  + Optional: ability to change an order’s status.
  + Optional: ability to create/edit/update/delete existing products and promotions.
* Optional: Administrator’s Product Management page
  + Optional: ability to change the status of a product’s stock (in stock or out of stock)

### Database Design

Table: User

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Note** |
| Id | Integer | Unique identifier |
| Username | Varchar |  |
| Password | depends | Stored as Bcrypt hash |
| Name | Varchar |  |
| Address | Varchar |  |
| Phone | Varchar |  |
| Email | Varchar |  |
| IsVerified | Boolean | **Optional**. Value is false until the user verifies their account via phone or email |
| Permission | Integer | **Optional**.  0 = customer  1 = administrator |

Table: Product

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Note** |
| Id | Integer | Unique identifier |
| Name | Varchar |  |
| Description | Text |  |
| Stock | Boolean | Value is false if the product is out of stock |
| Image | Text |  |
| Price | Integer |  |
| Type | String |  |

Table: Discount

A discount will have effect to all products.

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Datatype** | **Note** |
| Id | Integer | Unique identifier |
| Name | Text |  |
| DiscountAmount | Integer | Store the decimal value of the discount amount |
| StartDate | DateTime |  |
| EndDate | DateTime |  |

Table: Order

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Datatype** | **Note** |
| Id | Integer | Unique identifier |
| UserId | Integer |  |
| Price | Integer |  |
| OrderDate | DateTime |  |
| Status | Varchar | Pending/Processing/Shipping/Delivered/Cancelled |
| Address | String |  |
| Phone | String |  |

Table: OrderContent

|  |  |  |
| --- | --- | --- |
| **Column** | **Datatype** | **Note** |
| OrderId | Integer |  |
| ProductId | Integer |  |
| Quantity | Integer |  |
| Price | Integer |  |
| Size | Integer | 1 = Sm  2 = M  3 = L |
| Customizable | Integer | 1 = normal  2 = sweetness  3 = milk |

### Database Diagram

Diagram

Description automatically generated

## Login (CSO-36)

The process of logging into an account is performed as follow:

Diagram

Description automatically generated

The user will be have access to their Dashboard and all their permissions on the website once they log in successfully. They can do this in the **/login.html** page, which is accessible anywhere on the navigation bar.

Depends on the user’s type, certain permissions will be limited to them.

## Logout (CSO-37)

Once logging out, the user’s unique credentials that were created during the login process for authenticating purpose will be released, and they will become anonymous, this function is available anywhere on the navigation bar when a user is logged in.

## Signup (CSO-11)

The process when signing up is as follow. The user can do this in the **/signup.html** page, where they will be required to fill in their username, password and contact information. Upon submitting, their account will be created on the database, and they would be prompted to enter a unique code that the server generates to verify their account via either their provided phone number or email.

Diagram

Description automatically generated

Sprint 2

## Manage Products (CSO-18/CSO-8)

Only administrators have access to this feature.

This task is separated into 4 subtasks:

* Update Product (CSO-51)
* Read Product (CSO-52)
* Create Product (CSO-53)
* Delete Product (CSO-54)

Where any operation above will directly affect rows inside the Products table in the database. All products will be fetched into a list view, specific functionality to create a new product, delete an existing product and read a product’s details are available for the user to choose. The user will be prompted to input required information to add a product, or to modify a product’s information, their given data will be sent to the server as a form request, where the table rows will be affected.

Sprint 3

## Manage Orders (CSO-6)

* Manage state of an order (CSO-28): This tasks is separated into 2 aspects:
  + Any registered user can view their own order history, they can cancel any of their orders that are still in pending status at the moment of request.
  + Administrators can view all the orders and can modify any order’s status other than cancelling one. When any other isn’t in pending status, it can not be cancelled.
* Cancel and change details of an order (CSO-16)
  + A registered user can cancel any of their own order as long as it’s still in pending status.
  + While an order of a registered user is still pending, the user can change the contents of their order, consist of the information they gave during the checkout process, and the cart’s content.

## Manage Users (CSO-29/CSO-9)

* CRUD user information (CSO-29): This feature is implemented in the same logic as the Manage Product task.
* Get user’s information (CSO-41): This feature loads an user’s most of their given information during signup on a webpage, where they can be seen in their Dashboard page, an administrator can also see these information from the Manage Order menu.
* Get user’s purchase history (CSO-42): Any user can view their own order history in their Dashboard page. An administrator can also view this information in the Manage Order menu.

## Cart (CSO-86/CSO-43)

* Add to cart (CSO-62)
* Delete from cart (CSO-63)
* Update cart (CSO-64)

All of the tasks above works with Laravel’s Session. The cart contains information of products the user chose to buy from the store (it’s name, unique identifier, image, price, quantity, size and customizable) in the form of an array. Data sent to the server as form requests will then be compared to a temporary instance of the cart content, this temporary instance will then be added in new item, update an existing item or delete an existing item inside it, and then be assigned as the new cart Session.

Sprint 4

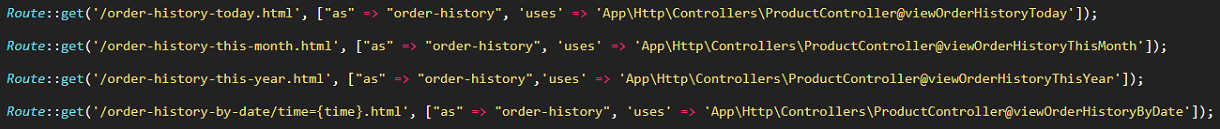
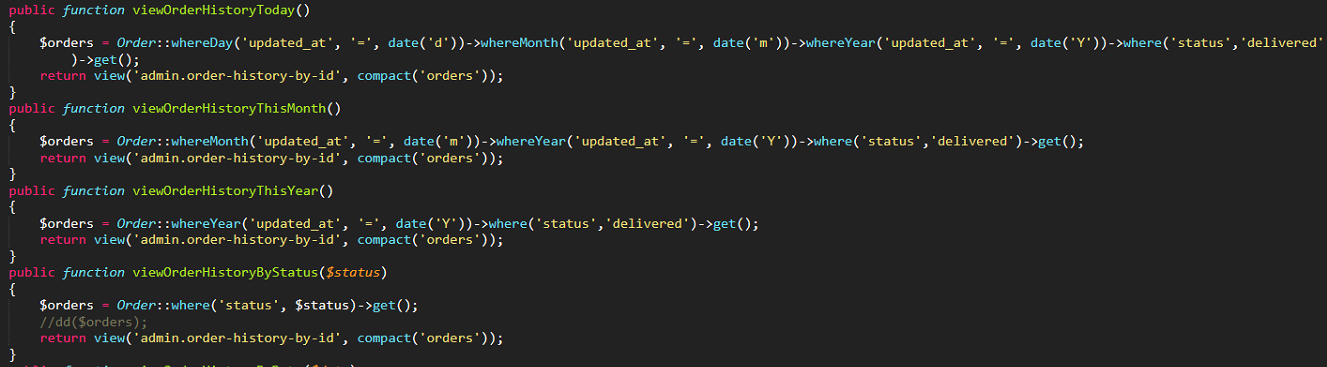
## Checkout (CSO-87)

* Pickup method (CSO-45): this functionality is implemented in the Checkout page, where the user can choose to either:
  + Pickup their order at the store
  + Pickup their order at their address (given during registration)
  + Pickup their order at an address given
* Payment direct (CSO-48): After the user confirm their checkout, the order will then be saved in the database in the Order table and all items bought will be saved in the Ordercontent table. Hypothetically, the user will pay once they have gotten their ordered goods.

## Show list product (CSO-52)

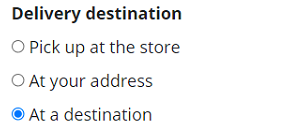
* Show list product contains 2 type:
  + Show all products and product type – products site (CSO-25)
  + Show a product details – product site (CSO-17)
* In products site, user can view the whole product listed in database or listed by a type. By click on a product, products site will redirect to product site’s link with ID as required parameter to query the exact item from database. By click on a type, products site will redirect to products site’s link with type as required parameter to query items with the matched type from database.
* In product site, user can view all product’s details and be able to add the current viewed product to Cart(CSO-86/43)

## Manage Order History (CSO-20)

* Show list product contains 5 types:
  + Function see full purchase history by exact date (extended)
  + Function see full purchase history by day (CSO-55)
  + Function see full purchase history by month (CSO-57)
  + Function see full purchase history by year (CSO-58)
* Route:   
  
* Controller:  
    
  Functions will return datas based on system current date time or input date.

## Shipping method (CSO-14)

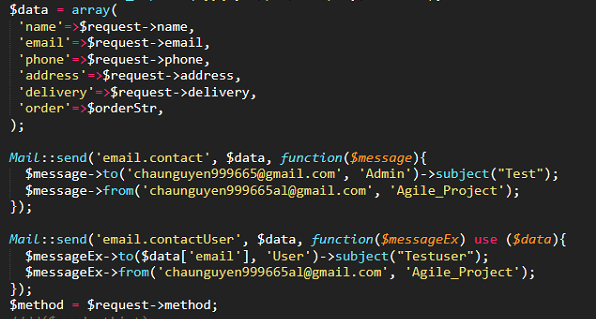
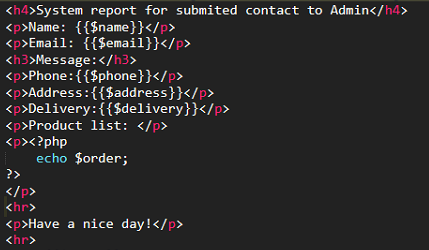
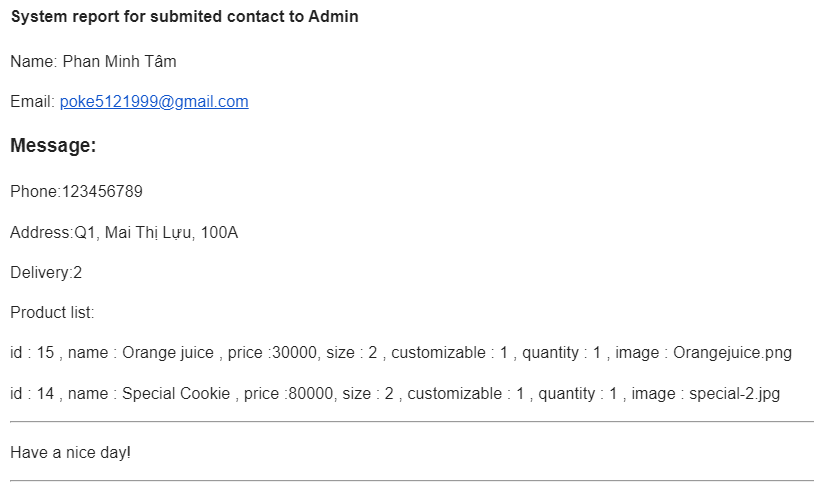
* By filling into the checkout form, user can tell if they want to pick up or ship their order:



## Payment method (CSO-15)

* Payment method contains 2 types:
  + Direct Payment (CSO-48)
  + Paypal Payment (CSO-50)  
    Route:  
      
    Controller:  
      
    

## Send Email system (CSO-93)

* Upon user send an order through system, an email will be sent to system Admin and user to notify that system has received the order informations.  
  Route:  
    
  Controller:  
    
  Template:  
   

## Analize (CSO-94)

* Show chart data of product and bill with time
  + Product Chart Tody
  + Product Chart This Month
  + Product Chart This Year
  + Bill day Per Week
  + Bill Week Per Month
  + Bill Month Per year

Route:



Controller:





## Tasks

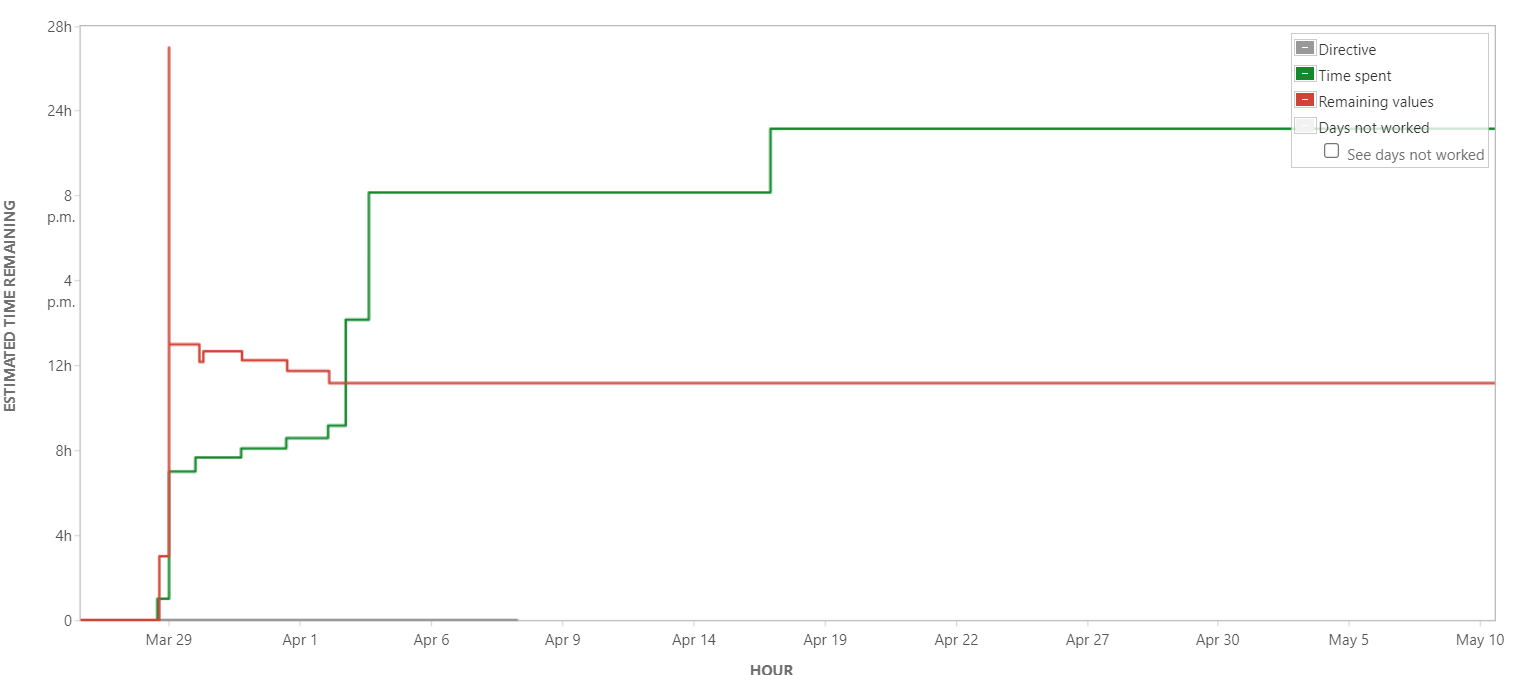
|  |  |  |
| --- | --- | --- |
| Epic | Task | Assignee |
| Authentication | Sign Up | Lê Trường Thanh |
| Register Function | Phan Huỳnh Văn Ngọc |
| Design Login Page | Lê Quang Huy |
| Login Function | Lê Trường Thanh |
| Logout Function | Lê Trường Thanh |
| Show List Product Of Store | Show List Product | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Show List Product UI | Lê Quang Huy |
| Manage Product | Manage Products | Lê Trường Thanh |
| Create Product | Lê Trường Thanh |
| Read Product | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Update Product | Lê Trường Thanh |
| Delete Product | Phan Huỳnh Văn Ngọc |
| Get Product List | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Product List UI | Lê Quang Huy |
| Product UI | Lê Quang Huy |
| User information, Order's state and history. | Design UserInfor-page(FE) | Nguyễn Thái Sang |
| Function see state of the order | Lê Trường Thanh |
| Manage state of the order | Lê Trường Thanh |
| Manage Order | Manage order history | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Function see full purchase history by day | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Function see full purchase history by week | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Function see full purchase history by month | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Function see full purchase history by year | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Cancel and change Order | Lê Trường Thanh |
| Function cancel order | Lê Trường Thanh |
| Function change order | Lê Trường Thanh |
| Design Manage Order History UI | Lê Quang Huy |
| Manage User | Get UserInfor UI | Lê Quang Huy |
| Design Register-page UI | Lê Quang Huy |
| Manage User | Lê Trường Thanh |
| Get user information list | Lê Trường Thanh |
| Get purchase history list | Lê Trường Thanh |
| Design Manage User UI | Lê Quang Huy |
| Cart | Cart UI | Lê Quang Huy |
| Cart | Lê Trường Thanh |
| Add To Cart | Lê Trường Thanh |
| Update Cart | Lê Trường Thanh |
| Delete from Cart | Lê Trường Thanh |
| CheckOut | Shipping Method | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Payment Method | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Checkout UI | Lê Quang Huy |
| External Payment UI | Lê Quang Huy |
| Pickup method | Lê Trường Thanh |
| Delivery method | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Payment Direct | Lê Trường Thanh |
| Payment by Paypal | Phan Huỳnh Văn Ngọc |
| Analyze and Graph | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Send Email system | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Test out system | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Create demo data | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu |
| Delayed tasks | Design Manage Order History UI | Lê Quang Huy |
| Design Manage User UI | Lê Quang Huy |

# Part 2: Sprints

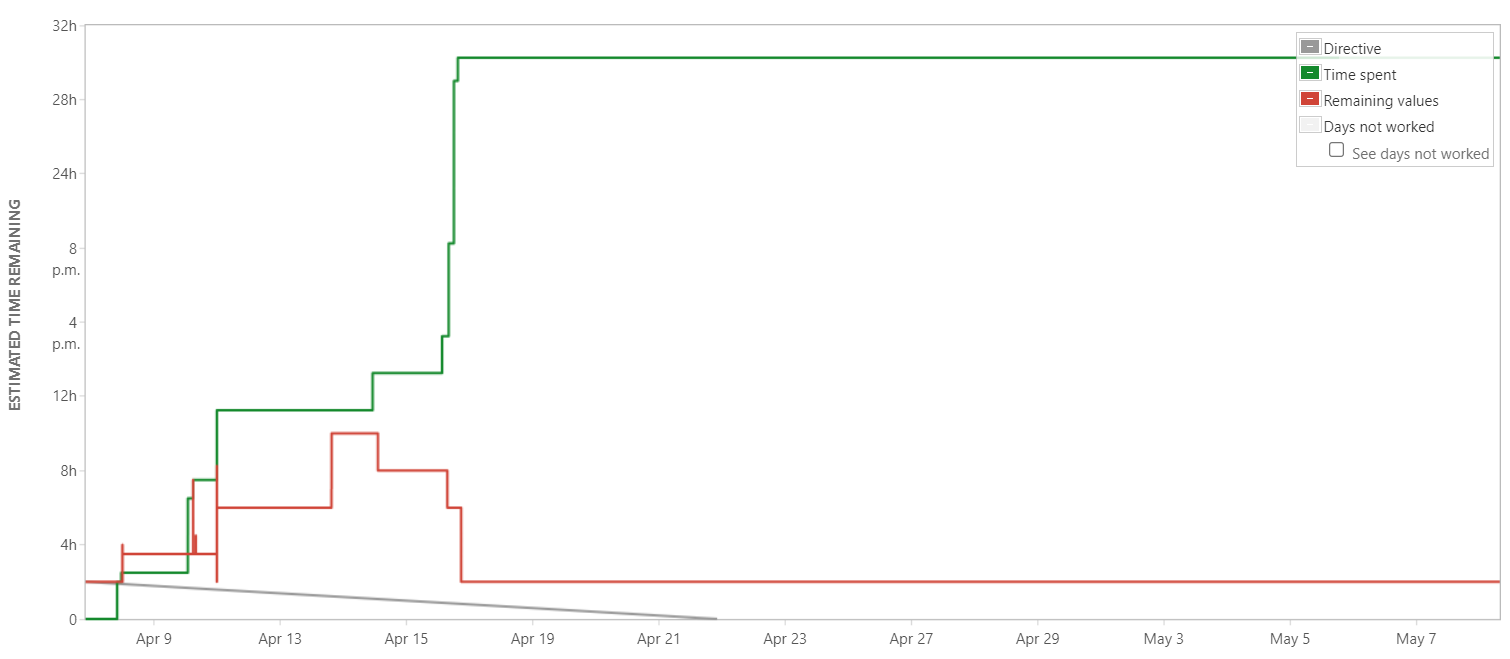
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint | Epic | Task | Assignee | Status | Estimated time |
| 1 | Authentication | Sign Up | Lê Trường Thanh | Done | 3 h |
| Register Function | Phan Huỳnh Văn Ngọc | Done | 3 h |
| Design Login Page | Lê Quang Huy | Done | 4 h |
| Login Function | Lê Trường Thanh | Done | 3 h |
| Logout Function | Lê Trường Thanh | Done | 3 h |
| Show List Product Of Store | Show List Product | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Show List Product UI | Lê Quang Huy | Done | 2 h |
| 2 | Manage Product | Manage Products | Lê Trường Thanh | Done | 5 h |
| Create Product | Lê Trường Thanh | Done | 3 h |
| Read Product | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 3 h |
| Update Product | Lê Trường Thanh | Done | 3 h |
| Delete Product | Phan Huỳnh Văn Ngọc | Done | 2 h |
| Get Product List | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 5 h |
| Product List UI | Lê Quang Huy | Done | 4 h |
| Product UI | Lê Quang Huy | Done | 4 h |
| User information, Order's state and history. | Design UserInfor-page(FE) | Nguyễn Thái Sang | Done | 4 h |
| Function see state of the order | Lê Trường Thanh | Done | 2 h |
| Manage state of the order | Lê Trường Thanh | Done | 2 h |
| 3 | Manage Order | Manage order history | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Function see full purchase history by day | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Function see full purchase history by week | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Function see full purchase history by month | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Function see full purchase history by year | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Cancel and change Order | Lê Trường Thanh | Done | 2 h |
| Function cancel order | Lê Trường Thanh | Done | 2 h |
| Function change order | Lê Trường Thanh | Done | 2 h |
| Design Manage Order History UI | Lê Quang Huy | Moved to Sprint 4 | 4 h |
| Manage User | Get UserInfor UI | Lê Quang Huy | Done | 2 h |
| Design Register-page UI | Lê Quang Huy | Done | 4 h |
| Manage User | Lê Trường Thanh | Done | 2 h |
| Get user information list | Lê Trường Thanh | Done | 2 h |
| Get purchase history list | Lê Trường Thanh | Done | 2 h |
| Design Manage User UI | Lê Quang Huy | Moved to Sprint 4 | 2 h |
| Cart | Cart UI | Lê Quang Huy | Done | 4 h |
| Cart | Lê Trường Thanh | Done | 2 h |
| Add To Cart | Lê Trường Thanh | Done | 2 h |
| Update Cart | Lê Trường Thanh | Done | 2 h |
| Delete from Cart | Lê Trường Thanh | Done | 2 h |
| 4 | CheckOut | Shipping Method | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 4 h |
| Payment Method | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Checkout UI | Lê Quang Huy | Done | 1 d |
| External Payment UI | Lê Quang Huy | To Do | 4 h |
| Pickup method | Lê Trường Thanh | Done | 1 h |
| Delivery method | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Payment Direct | Lê Trường Thanh | Done | 2 h |
| Payment by Paypal | Phan Huỳnh Văn Ngọc | Done | 3 d |
| Analyze and Graph | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 6 h |
| Send Email system | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Test out system | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 2 h |
| Create demo data | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu | Done | 6 h |
| Delayed tasks | Design Manage Order History UI | Lê Quang Huy | Done | [delayed task] |
| Design Manage User UI | Lê Quang Huy | Done | [delayed task] |
| Extra | Document | Report writing | Phan Minh Tâm + Nguyễn Hoàng Kỳ Châu  Lê Trường Thanh  Phan Huỳnh Văn Ngọc | Done | None |
| Story and tasks deploy |  | Lê Quang Huy | Done | None |
| Extra tasks deploy |  | Nguyễn Hoàng Kỳ Châu | Done | None |

## Burndown chart

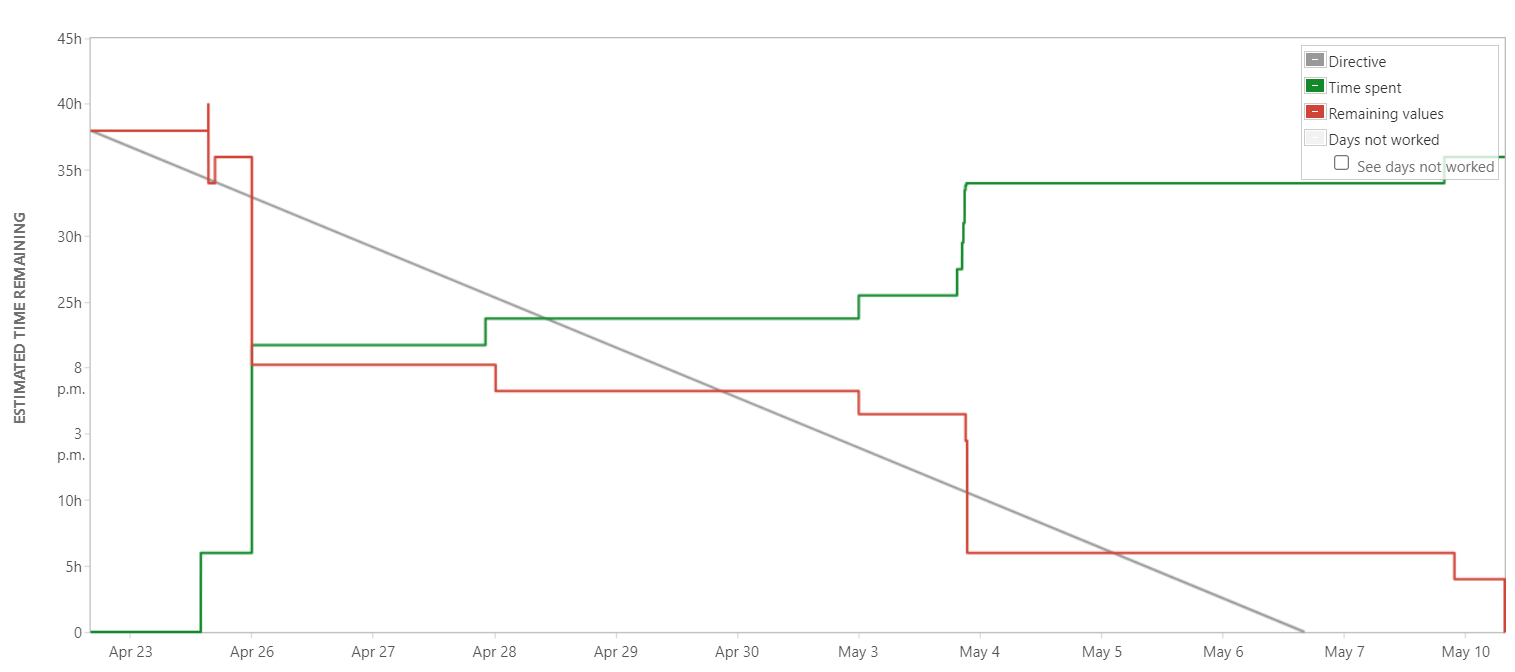
Sprint 1



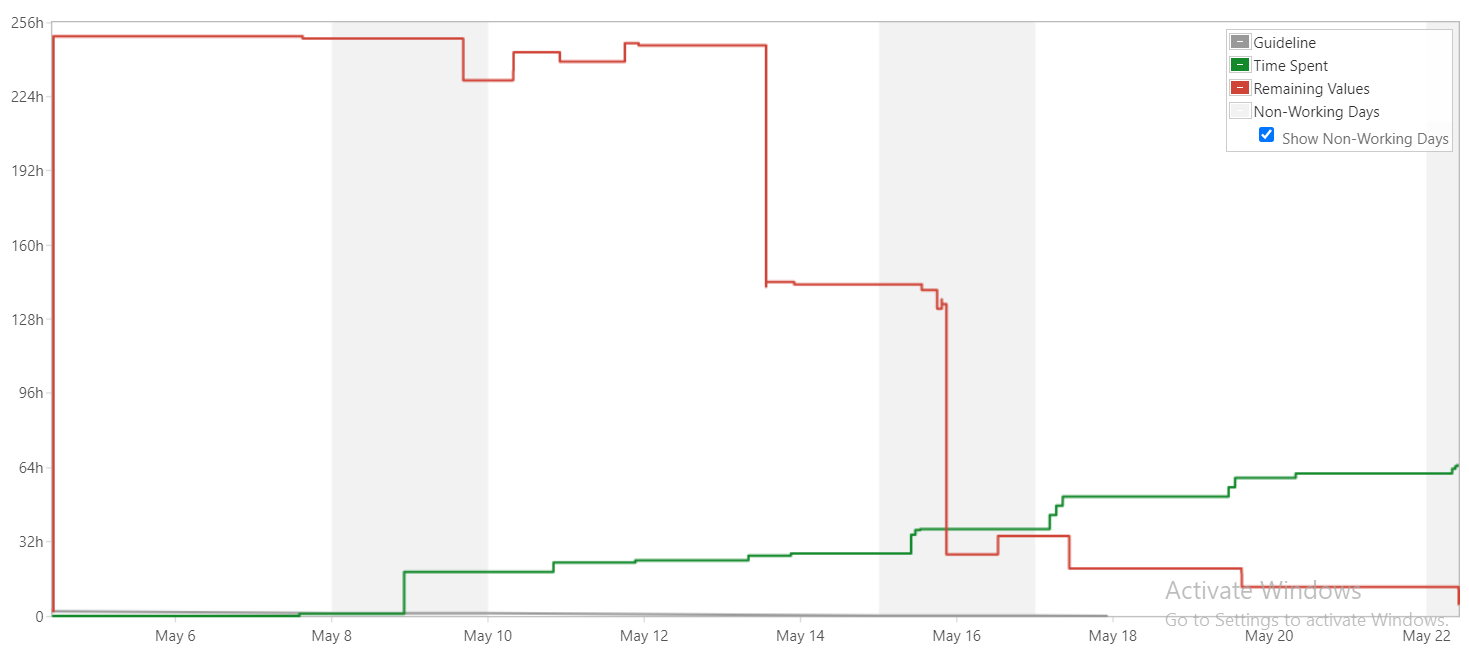
Sprint 2



Sprint 3

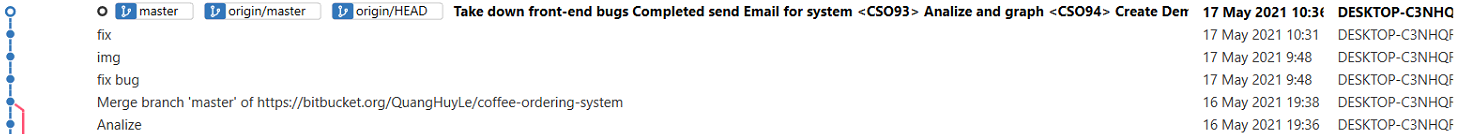
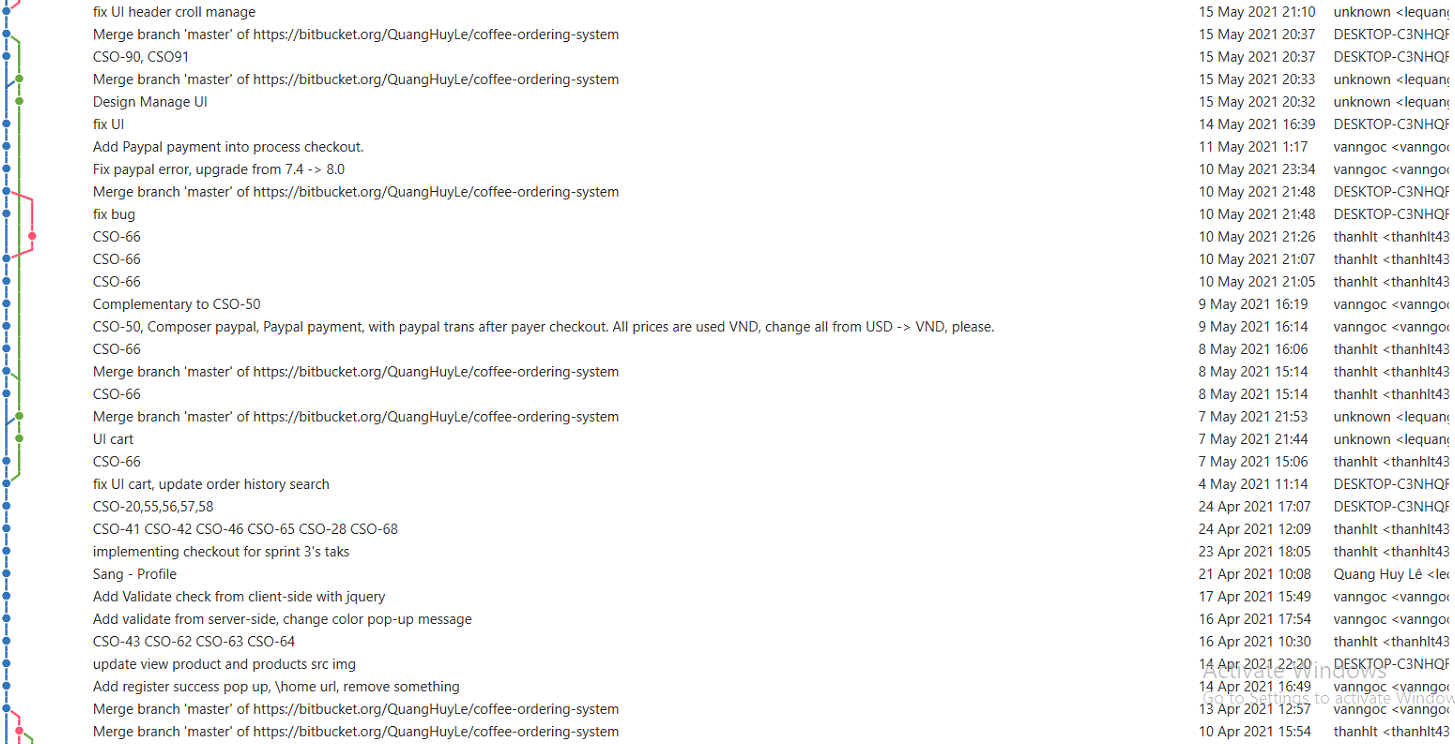


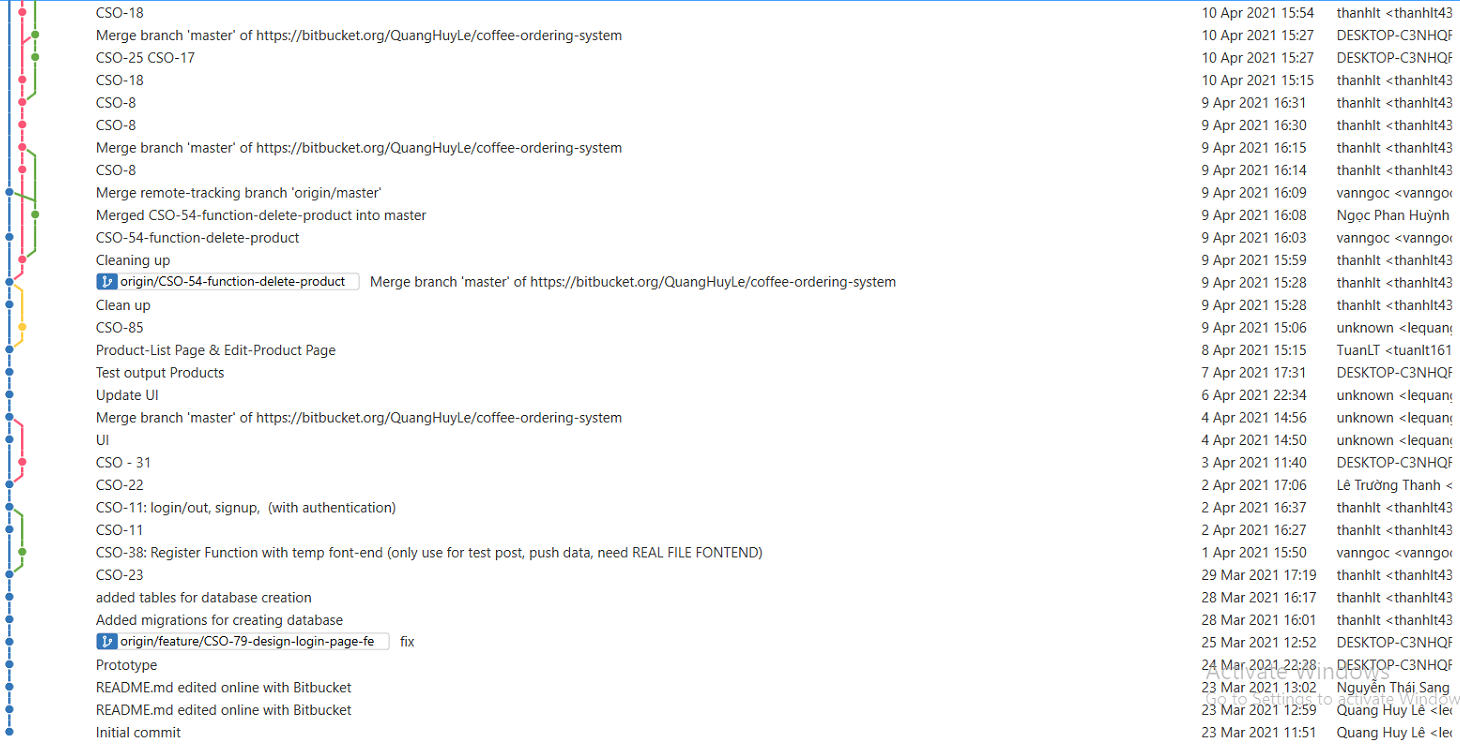
Sprint 4



Total time spent: 144hours

Commit list





# Part 3: Total evaluation

Within the implementation period team have completed most of designed fucntions and some other extended functions.