HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

**FACULTY OF COMPUTER SCIENCE AND ENGINEERING**

****

PROJECT REPORT

PRACTICE ON SOFTWARE ENGINEERING

Group PTSD\_L01

POS SYSTEM

Lecturer: Assoc. Prof. Quan Thanh Tho

|  |  |
| --- | --- |
| Huynh Ba Dat | 1811854 |
| Mai Van Hung | 1812491 |
| Vu Minh Duong | 1810885 |
| Tran Viet Tuan Kiet | 1913887 |
| Hoang Cong Son | 1911975 |

# Task 1: Requirement elicitation

## **Task 1.1**

### Context

* Point of sale (POS) or point of purchase is the time and place where a retail transaction is completed. In restaurant business, POS systems often include table reservation, ordering food, alerts, billing, credit card processing and customer management. Even before the COVID-19 crisis, POS systems had gained traction across the industry.
* During the coronavirus pandemic, restaurants face greater peril than ever. Such systems are expected to increase business intelligence, reduce wasted effort and opportunity to scale to a large business. Moreover, the systems should support take-away options.

### Relevant stakeholders

* Restaurant manager
* Customer
* Clerk
* Developer

### Features

* Customer:

+ Table reservation

+ Skim Menu

+ Select a Meal /Select Recommendation

+ Placing an Order

+ Sending Feedback

- Restaurant:

+ Customer management

+ Confirm Order

+ Processing Order

+ Recording Order

+ Recording Transaction

+ Sending Feeback

### 4. Scope

- Visa/MasterCard only

- 300 orders per day

- Web/QR code

- Mobile device/tablet device/computer/laptop

- Use in multiple restaurants

## **Task 1.2**

### 1. Funtional requirements:

* Skim Menu: The website displays a list of products for customers to choose from

- Select a Meal /Select Recommendation: Customers can choose dishes by manipulating the Menu page directly through their device

* Placing an Order: The selected dishes are displayed in “Your Cart” with information and prices to pay
* Sending Feeback: Customers can leave feedback about food and service through the web

- Confirm Order: Clerk confirm customer's menu

* Recording Order: POS terminal and Clerk records the customer's order
* Processing Order: POS terminal processes order and then transmits the data to the kitchen
* Recording Transaction: POS terminal records customer information in transaction history
* Receiving Payment: POS terminal receives payment from customers via VISA or Card

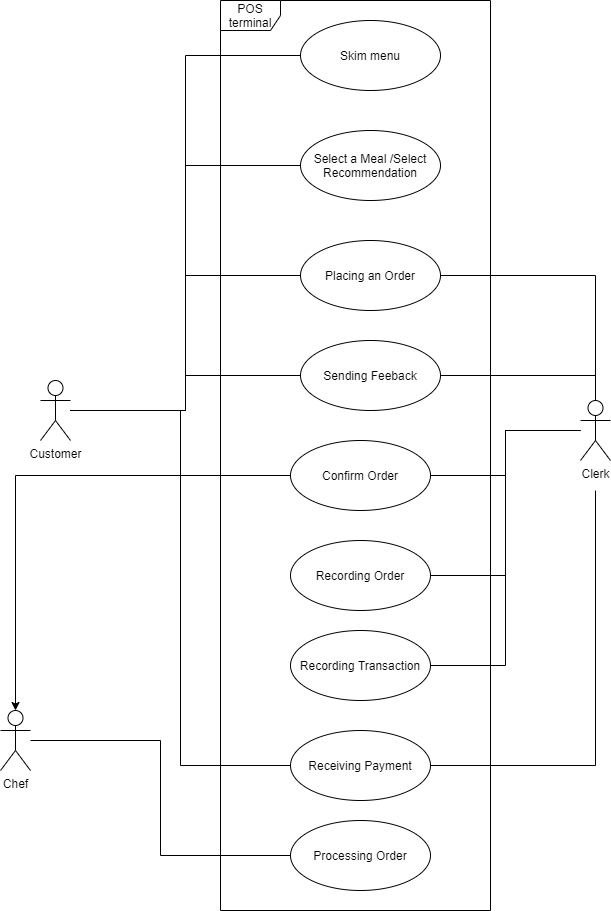
### 2. Non-functional requirements:

* Non-direct contact between Clerks and Customers
* Using Web technology and QR code
* Transactions is about 300 orders per day
* The system should be extendable to use in multiple restaurants in the future
* The system should be usable from a mobile device, a tablet device or a normal computer/laptop
* Time loading belows 2s
* Website work on multiple browsers
* Design easy-to-use navigation

### 3. Actor:

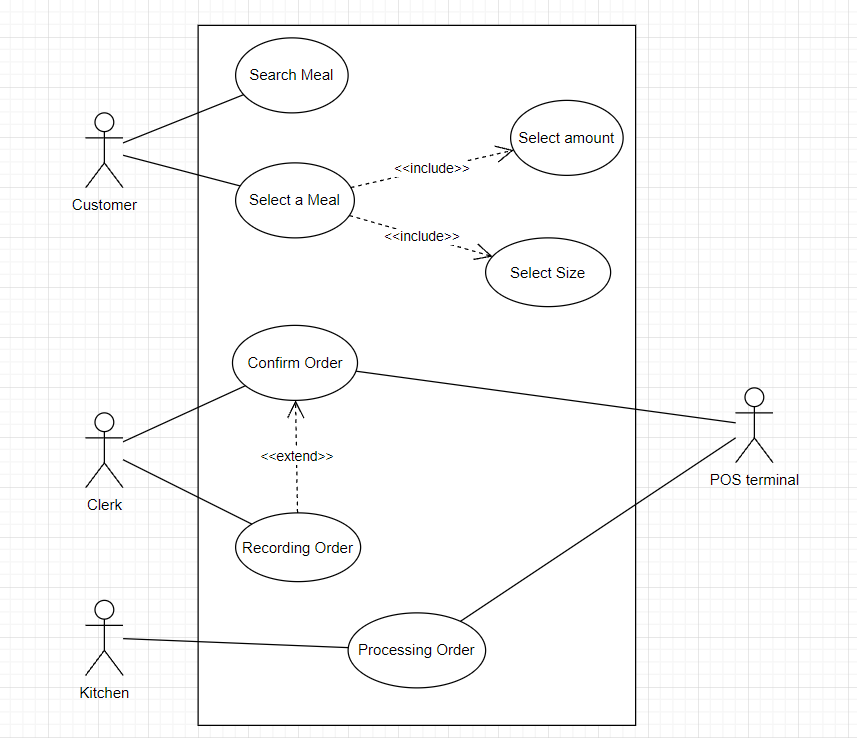
* Clerk
* Customer
* Kitchen
* POS terminal

### 4. Use-case diagram



## **Task 1.3**

- Use-case Description: Food ordering:



|  |  |
| --- | --- |
| Name | Search Meal |
| Actor | Customers |
| Description | Customers can choose a product group (eg Cupcake, Seafood, Juice, etc.) and find the item they like. |
| Preconditions | Customers need to visit the website. |
| Normal flow | 1. Customers access the website by scanning the QR code or URL.  2. Customer selects a group of items.  3. Customers surf to find the food they want. |
| Exceptions | Exception at step 3: Customer could not find the item he was looking for. |
| Alternative flow | At step 3, Customers can ask staff for more details. |

|  |  |
| --- | --- |
| Name | Select Meal |
| Actor | Customers |
| Description | Customers can choose dishes (eg: Pancake, Seafood, Juice,...) and set order details. |
| Preconditions | Customers need to visit the website. |
| Normal flow | 1. Customers access the website by scanning the QR code or URL  2. Customers click on the food they want  3. The web page displays a window containing information about the dish  4. Customers read information about the dish  5. Customers choose the size and quantity and press the buy button 6. The website returns to the menu screen and displays the Your Cart window containing the selected dishes in the right corner of the screen  7. Customers choose other dishes |
| Exceptions | Exception at step 5: The customer does not want to add the currently displayed item to the cart |
| Alternative flow | At step 4, Customers can press the “back” button to return to the menu screen and continue to order |

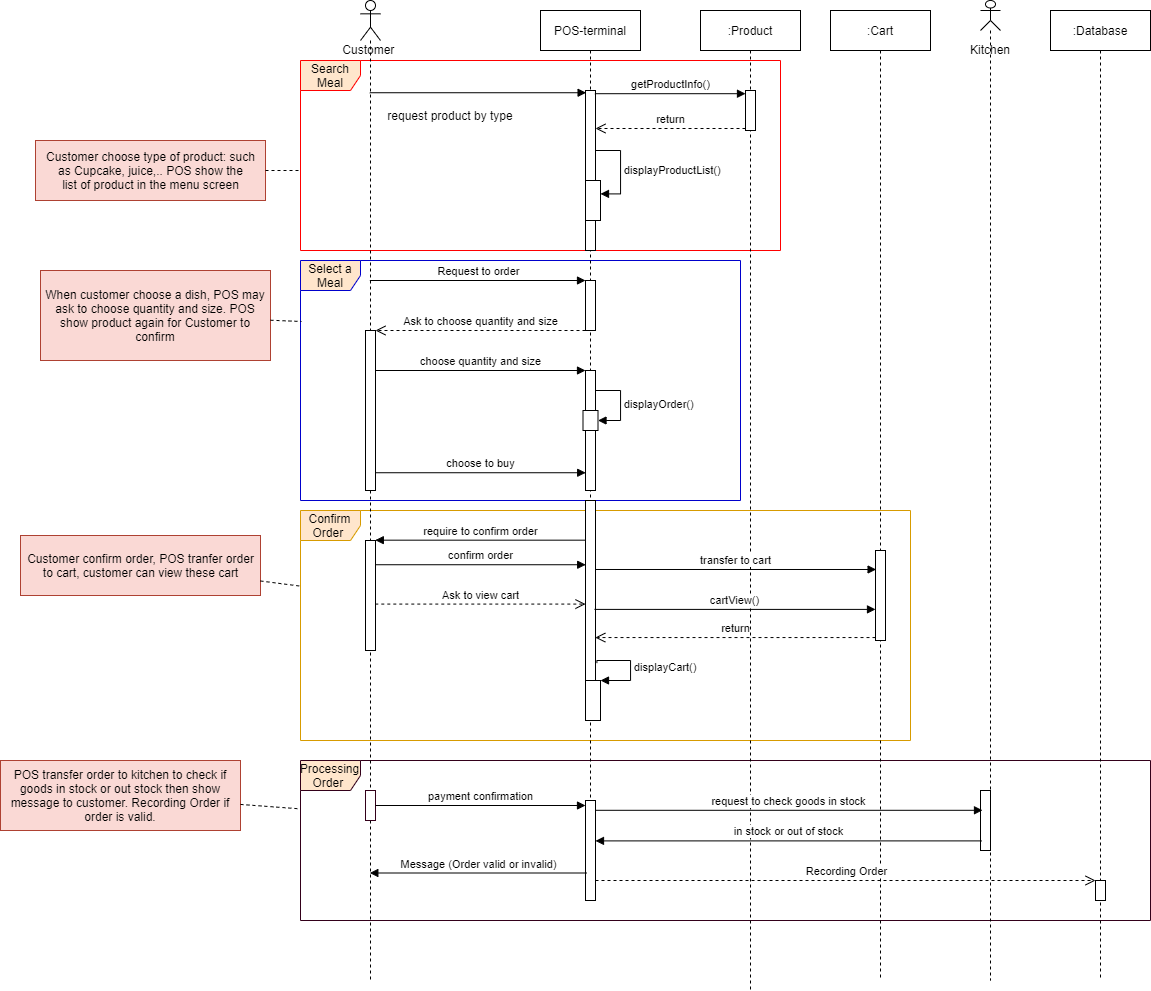
|  |  |
| --- | --- |
| Name | Confirm Order |
| Actor | Clerk |
| Description | Clerk confirm customer's order |
| Preconditions | Customers need to order |
| Normal flow | 1. The clerk confirms with the customer the number of dishes, the number of products per item, the amount to be paid through the customer's order software.  2. Customer confirms order  3. Valid order confirmation system.  4. Clerk records orders |
| Exceptions | Exception at step 3: Invalid order response system because out of stock |
| Alternative flow | At step 3, Clerk responds to customers to cancel or choose another item |

|  |  |
| --- | --- |
| Name | Processing Order |
| Actor | POS, Kitchen |
| Description | The chef checks the amount of goods in stock and responds to valid or invalid orders through the system. |
| Preconditions | Employees must submit orders to the system |
| Normal flow | 1. POS sends customer's order to kitchen's order taking software  2. The kitchen checks the stock and gives feedback to POS  3. POS feedback via customer order software that is the order valid or invalid |
| Exceptions |  |
| Alternative flow |  |

# Task 2: System modeling

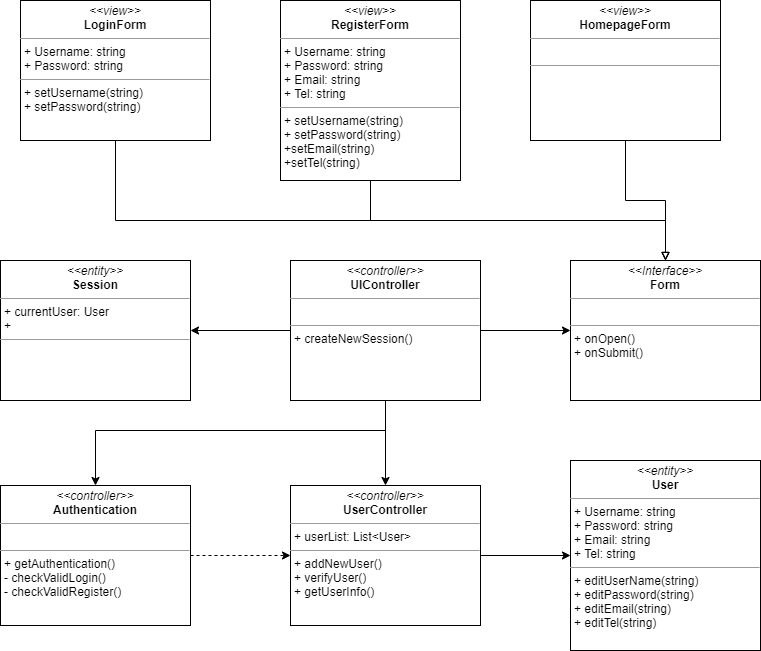
## **Task 2.1**

## **Task 2.2**

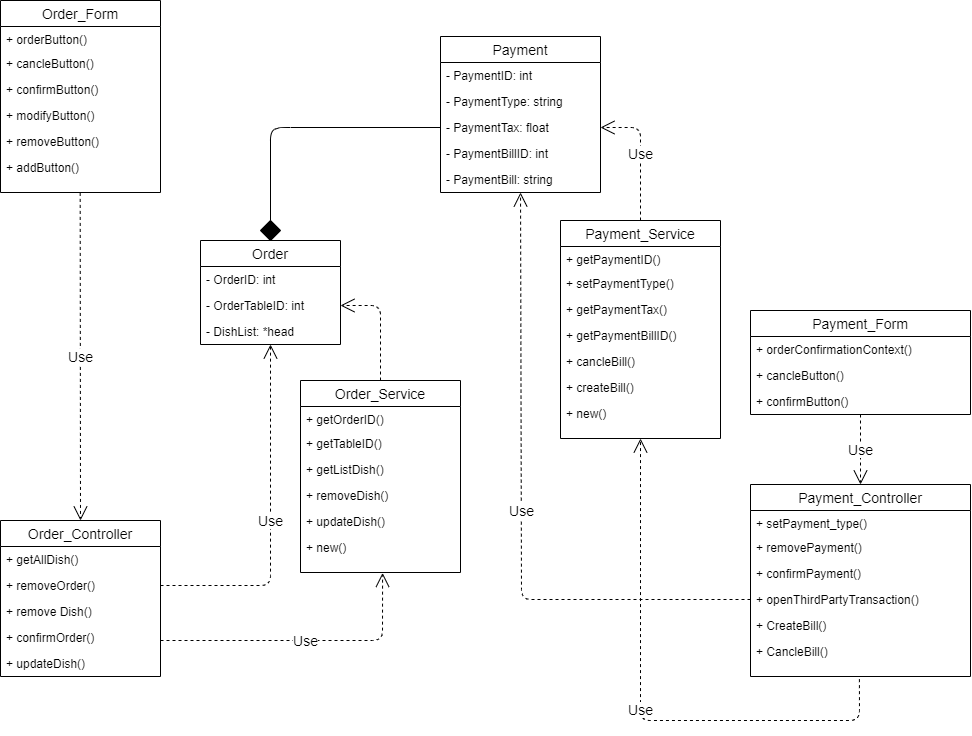


## **Task 2.3**

### Homepage:



### Order:



# Task 3: Architecture design

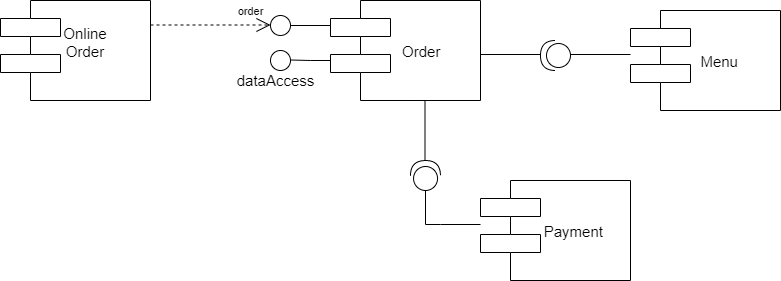
## **Task 3.1**

### Deployment view

## **Task 3.2**

### Implementation view

#### Order:

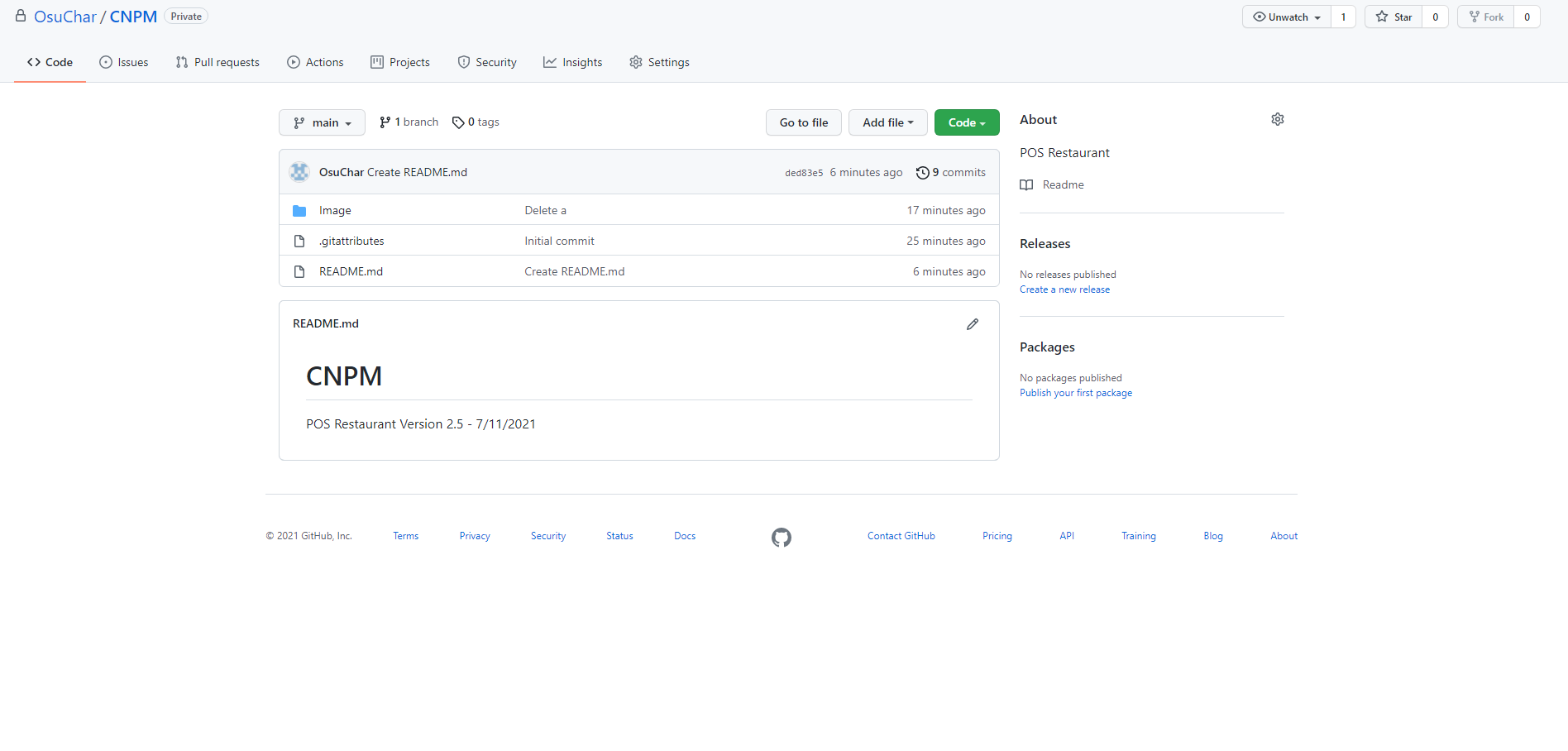


#### Payment:

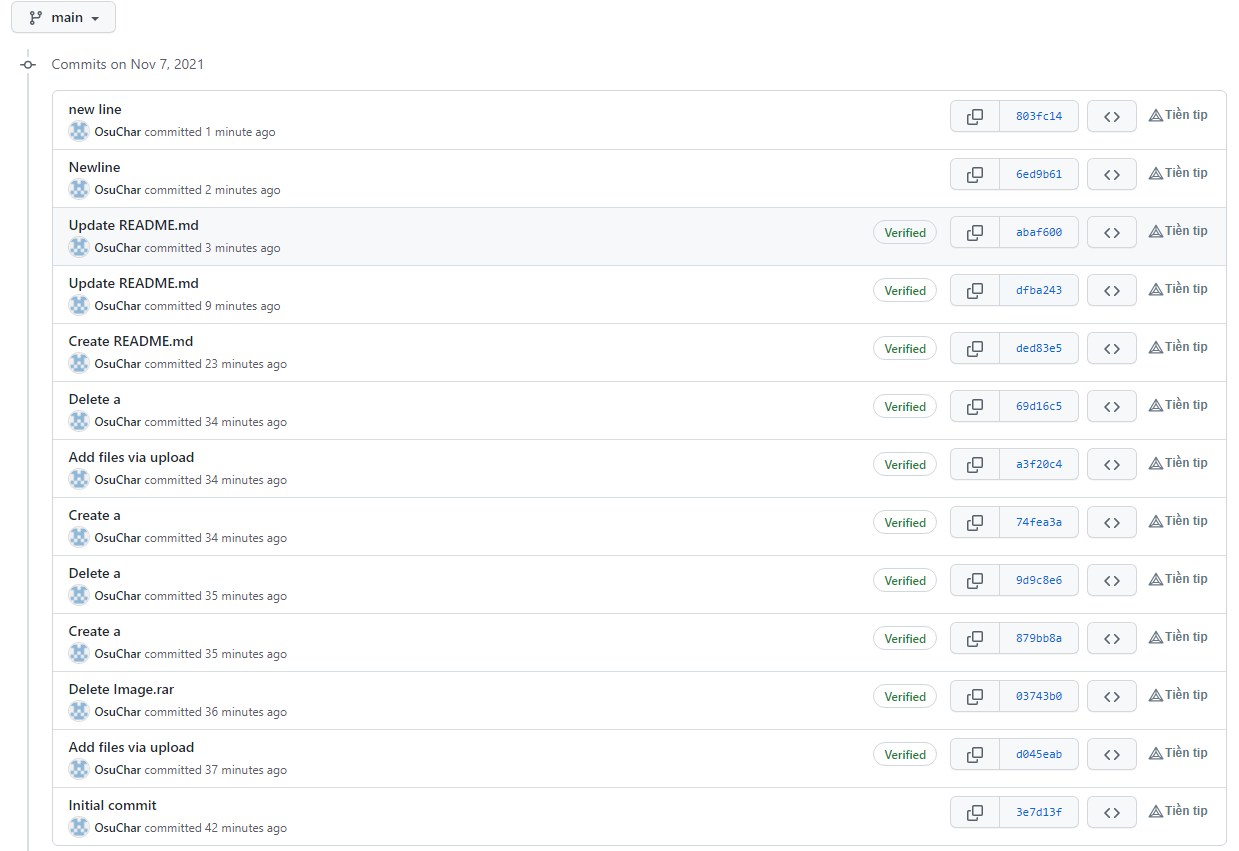
# Task 4: Implementation – Sprint 1

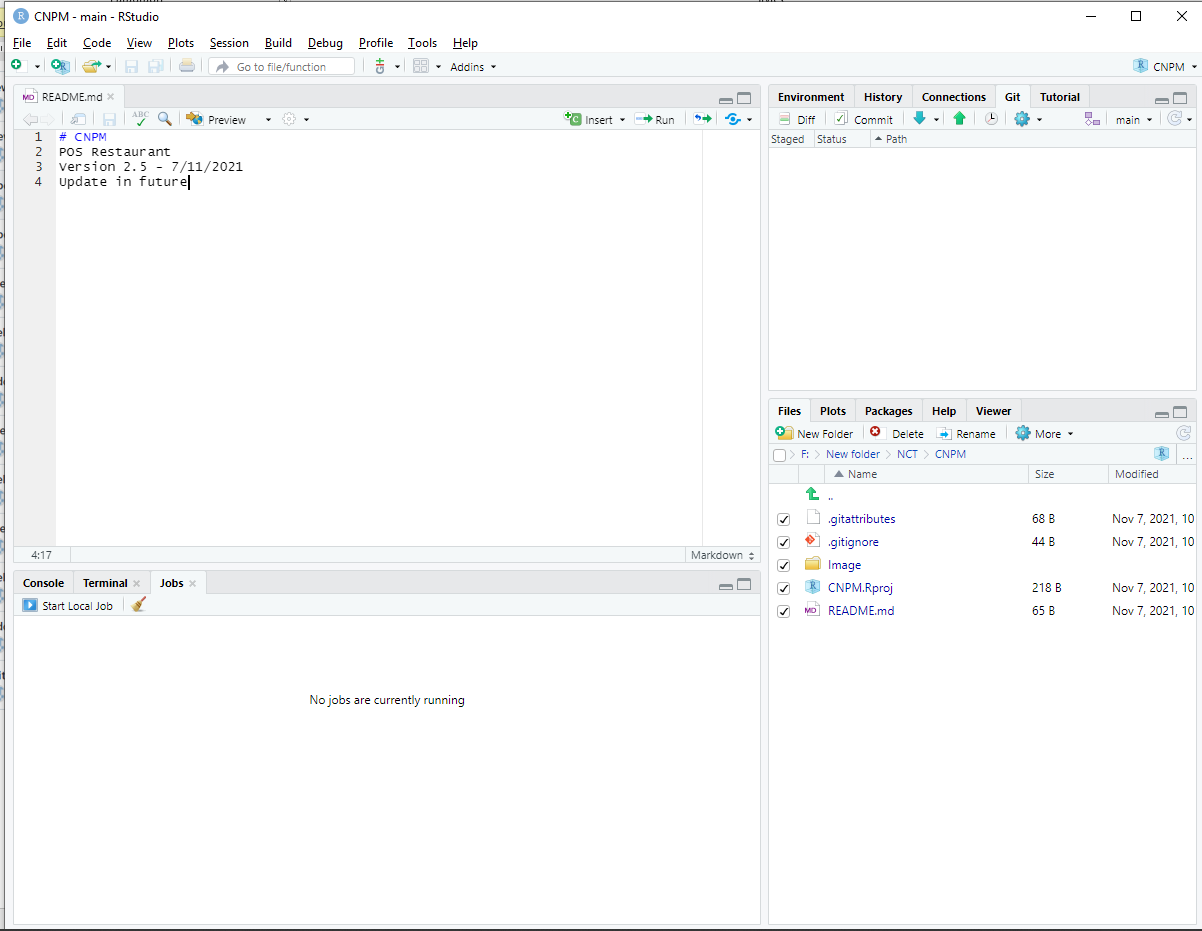
## **Task 4.1**

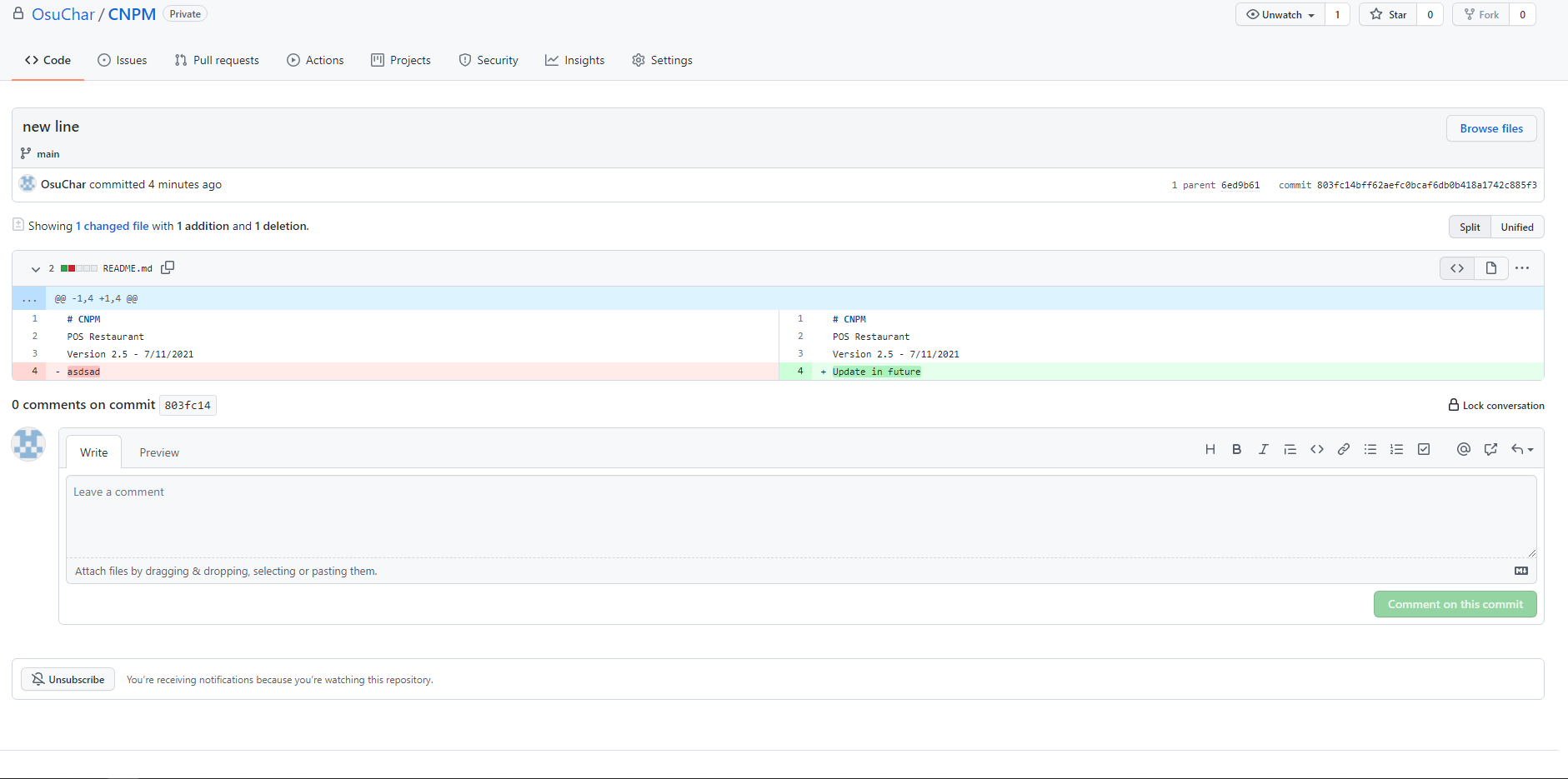
Link: https://github.com/OsuChar/CNPM



## **Task 4.2**







## **Task 4.3**

