QUẢN LÝ QUI TRÌNH PHẦN MỀM

**QUY ƯỚC MÃ NGUỒN**

**GV PHỤ TRÁCH:**

**Thầy Ngô Ngọc Đăng khoa**

**Thầy Nguyễn Đức Huy**

**NHÓM J**

Phiên bản 1.0 Ngày 22/06/2021



Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

**Bảng phân công chính**

|  |  |
| --- | --- |
| Chức vụ | Người chịu trách nhiệm chính |
| Project Manager | Quách Hoàng Minh |
| DEV | Phan Nguyễn Trọng Đại |
| DES | Nguyễn Bảo Nguyên |

**Bảng phân công chi tiết**

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ và tên | Nhiệm vụ | Đánh giá công việc  (0 – 10) |
| 1859005 | Phan Nguyễn Trọng Đại |  | 10 |
| 1859033 | Quách Hoàng Minh |  | 10 |
| 1859035 | Nguyễn Bảo Nguyên |  | 10 |

# LỊCH SỬ THAY ĐỔI

|  |  |  |  |
| --- | --- | --- | --- |
| **Phiên bản** | **Tác giả** | **Ngày** | **Lý do** |
| *1.0* | Phan Nguyễn Trọng Đại | 22/06/2021 | Soạn thảo văn bản |

Table of Contents

[LỊCH SỬ THAY ĐỔI 3](#_Toc75265125)

[a. CÁC NGÔN NGỮ SỬ DỤNG 5](#_Toc75265126)

[1. Thiết kế module: 5](#_Toc75265127)

[2. Cách tổ chức source code: 5](#_Toc75265128)

[3. Quy cách code: 5](#_Toc75265129)

[a. Quy cách đặt tên: 5](#_Toc75265130)

[b. Quy cách viết code: 5](#_Toc75265131)

[4. Handle Error và Logging: 7](#_Toc75265132)

[b. Handle Error: 7](#_Toc75265133)

[c. Logging: 7](#_Toc75265134)

[5. Tool và IDE: 7](#_Toc75265135)

# Các ngôn ngữ sử dụng

|  |  |
| --- | --- |
| Tên | Vai trò |
| Javascript | Cài đặt chính |
| ReactJS | Client |

# Thiết kế module:

Trong dự án Case-Study của nhóm J sẽ được chia thành các module riêng biệt, mỗi module của mỗi components sẽ thực hiện đúng chức năng của nó. Sẽ không nói rõ chi tiết tại đây, tuy nhiên Code sẽ đơn giản, rõ ràng, tách biệt và liên kết khi cần thiết.

# Cách tổ chức source code:

Các files có cùng “tính chất” sẽ được tổ chức trong cùng 1 thư mục tương ứng. Ví dụ: các file thực hiện các chức năng của đồ án sẽ để trong folder “components”.

# Quy cách code:

## Quy cách đặt tên:

* Đặt tên biến/hàm phải đi liền với ý nghĩa sử dụng.  
  Ví dụ: const characterLimit = “500”; các kí tự được nhập trong input form sẽ có giới hạn là người dùng được nhập vào tối đa 500 từ…
* Tên biến và hàm nên đặt theo camelCase (javascript)  
  Ví dụ: characterLimit
* Tên hàm nên là động từ và tên biến là danh từ  
  Ví dụ: var error, function setError…
* Các biến hoặc hàm trả về kiểu boolean nên dùng “is”, “can”, “have”,… đứng trước tên biến/hàm  
  Ví dụ: isValid …
* Các biến tạm, hay đặt tên rõ ràng, ko sử dụng temp.
* Các interface thì phải có chữ “I” phía trước.

## Quy cách viết code:

* Giữa các tên biến, values, properties, và objects phải có các kí tự “=”, “:”,.. phải có khoảng trắng  
  Ví dụ: var error = false;  
  var data = { id: nanoid() };
* Luôn nhớ sử dụng dấu chấm phẩy “;”, cả trong trường hợp không bị lỗi nếu không sử dụng dấu chấm phẩy thì cũng nên sử dụng. (trong javascript)
* Ngắt dòng code khi cần thiết, để dễ nhìn hơn.
* Thống nhất 1 cách đóng 1 code block
* Ví dụ: trong javascript  
  if ( myFunction() ) {  
   // Expressions  
  } else if ( ( a &amp;&amp; b ) || c ) {  
   // Expressions  
  } else {  
   // Expressions  
  }
* Comment code: comment ở đầu file để đưa ra overview, hoặc ở những đoạn code phức tạp, khó hiểu, không nhất thiết phải ở tất cả các dòng code
* Comment 1 dòng: sử dụng //
* Comment nhiều dòng: sử dụng /\* \*/
* Comment format trong React {\* \*}

# Handle Error và Logging:

## Handle Error:

* Lỗi cú pháp hầu hết có thể xảy ra trong giai đoạn phát triển ban đầu và là kết quả của sai cú pháp. Lỗi cú pháp có thể được bắt gặp và sửa chữa một cách dễ dàng khi chương trình được biên dịch để thực thi.
* Lỗi logic là kết quả của việc thực hiện logic không hợp lý. Phát hiện và sửa lỗi dựa trên specification, requirement và bug log từ tester.
* Lỗi runtime là các lỗi khi chúng ta không cân nhắc tất cả các trường hợp xảy ra. Thông thường sẽ sử dụng try catch trong code để handle.

## Logging:

Ghi log là để lưu lại quá trình thực thi của chương trình.

* Nếu ghi log cho lỗi (thường là lỗi runtime) thì khi chương trình có lỗi có thể nhìn được lỗi gì, ở đoạn code nào và dễ dàng trong việc tìm phương án xử lý. Nhiều trường hợp chương trình gặp lỗi trên môi trường production mà không thể tái hiện lại được, khi đó sẽ xem log để tìm nguyên nhân.
* Nếu ghi log cho quá trình sử dụng ứng dụng, sẽ biết user nào vào truy cập vào ứng dụng và thao tác gì, từ đó tìm được thủ phạm của những tác nhân gậy hại đến ứng dụng
* Nếu ghi log ở các đoạn code phức tạp, các tiến trình chạy ngầm, sẽ biết được trạng thái của tiến trình đó đang chạy như thế nào, biết được tốc độ xử lý của các nghiệp vụ phức tạp, có cơ sở để tìm ra các đoạn code gây cao tải cho hệ thống…

Các cách ghi log:

* Print ra màn hình
* Ghi ra file lưu trên ổ cứng
* Insert log vào 1 bảng trong cơ sở dữ liệu...

Tùy vào ngôn ngữ và công nghệ sử dụng mà có các phương pháp ghi log khác nhau. Ví dụ như React sẽ có react tool developer hỗ trợ cho việc ghi log.

# Tool và IDE:

Visual Studio Code, Postman, ...