**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

🙞🕮🙜

Logo

Description automatically generated

**Đề tài:**

**XÂY DỰNG ỨNG DỤNG WEB**

**THI ONLINE AN TOÀN**

**SỬ DỤNG CÔNG NGHỆ MERN STACK**

**MÔN: KIỂM THỬ PHẦN MỀM**

**GIÁO VIÊN HƯỚNG DẪN**

**Th.S. Nguyễn Trần Thi Văn**

**SINH VIÊN THỰC HIỆN**

|  |  |
| --- | --- |
| Nguyễn Thị Minh Hoàng | 18110285 |
| Trần Ngọc Minh Thiện | 18110371 |
| Nguyễn Huỳnh Minh Tiến | 18110377 |
| Lê Thị Ngọc Yến | 18110402 |

**TPHCM, tháng 12 năm 2021**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN CÔNG NGHỆ PHẦN MỀM**

🙞🕮🙜

Logo

Description automatically generated

**Đề tài:**

**XÂY DỰNG ỨNG DỤNG WEB**

**THI ONLINE AN TOÀN**

**SỬ DỤNG CÔNG NGHỆ MERN STACK**

**MÔN: KIỂM THỬ PHẦN MỀM**

**GIÁO VIÊN HƯỚNG DẪN**

**Th.S. Nguyễn Trần Thi Văn**

**SINH VIÊN THỰC HIỆN**

|  |  |
| --- | --- |
| Nguyễn Thị Minh Hoàng | 18110285 |
| Trần Ngọc Minh Thiện | 18110371 |
| Nguyễn Huỳnh Minh Tiến | 18110377 |
| Lê Thị Ngọc Yến | 18110402 |

**TPHCM, tháng 12 năm 2021**

|  |  |
| --- | --- |
| **ĐH SƯ PHẠM KỸ THUẬT TP. HCM**  **KHOA CNTT**  \*\*\*\*\*\*\* | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh Phúc**  \*\*\*\*\*\*\* |

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên 1: **Nguyễn Thị Minh Hoàng** MSSV 1: **18110285**

Họ và tên Sinh viên 2: **Trần Ngọc Minh Thiện** MSSV 2: **18110371**

Họ và tên Sinh viên 3: **Nguyễn Huỳnh Minh Tiến** MSSV 2: **18110377**

Họ và tên Sinh viên 4: **Lê Thị Ngọc Yến** MSSV 1: **18110402**

Ngành: **Công nghệ Thông tin**

Tên đề tài: **Xây dựng ứng dụng web thi online an toàn sử dụng công nghệ MERN Stack**

Họ và tên Giáo viên hướng dẫn: **Th.S. Nguyễn Trần Thi Văn**

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đánh giá loại:

5. Điểm:

*Tp. Hồ Chí Minh, ngày tháng năm 2022*

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

# **LỜI CẢM ƠN**

*Lời đầu tiên nhóm xin phép được gửi lời cảm ơn chân thành và sâu sắc nhất đến với Khoa Công Nghệ Thông Tin – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh đã tạo điều kiện cho nhóm chúng em được học tập, phát triển nền tảng kiến thức sâu sắc và thực hiện đề tài này.*

*Bên cạnh đó nhóm chúng em xin gửi đến thầy Nguyễn Trần Thi Văn lời cảm ơn sâu sắc nhất. Trải qua một quá trình dài học tập và thực hiện đề tài trong thời gian qua. Thầy đã tận tâm chỉ bảo nhiệt tình nhóm chúng em trong suốt quá trình từ lúc bắt đầu cũng như kết thúc đề tài này.*

*Nhờ có những nền tảng kiến thức chuyên ngành vững chắc cộng thêm với những kinh nhiệm và yêu cầu thực tế ngoài xã hội thông qua việc học ở trường và thực tập ở các công ty. Tập thể các thầy cô Khoa Công Nghệ Thông Tin và đặc biệt thầy Nguyễn Trần Thi Văn đã tặng cho chúng em một khối lượng kiến thức và kinh nghiệm khổng lồ về chuyên ngành và công việc trong tương lai. Đặc biệt điều này đã giúp và thôi thúc chúng em hoàn thành được đề tài. Đây sẽ là hành trang vô cùng lớn của chúng em trước khi bước ra một cuộc sống mới.*

*Tuy nhiên lượng kiến thức là vô tận và với khả năng hạn hẹp chúng em đã rất cố gắng để hoàn thành một cách tốt nhất. Chính vì vậy việc xảy ra những thiếu sót là điều khó có thể tránh khỏi. Chúng em hi vọng nhận được sự góp ý tận tình của quý thầy (cô) qua đó chúng em có thể rút ra được bài học kinh nghiệm và hoàn thiện và cải thiện nâng cấp lại sản phẩm của mình một cách tốt nhất có thể.*

*Chúng em xin chân thành cảm ơn!*

|  |  |
| --- | --- |
| **Nhóm thực hiện** | |
| Nguyễn Thị Minh Hoàng | 18110285 |
| Trần Ngọc Minh Thiện | 18110371 |
| Nguyễn Huỳnh Minh Tiến | 18110377 |
| Lê Thị Ngọc Yến | 18110402 |

**MỤC LỤC**

[LỜI CẢM ƠN 4](#_Toc89260595)

[MỤC LỤC 5](#_Toc89260596)

[LỜI GIỚI THIỆU 7](#_Toc89260597)

[NỘI DUNG 8](#_Toc89260598)

[**1.** **ĐẶC TẢ VÀ MÔ HÌNH HÓA YÊU CẦU VÀ GIỚI THIỆU CÁC CHỨC NĂNG CỦA ỨNG DỤNG** 8](#_Toc89260599)

[**2.** **CƠ SỞ LÝ THUYẾT VỀ KIỂM THỬ** 9](#_Toc89260600)

[**2.1.** **Kiểm thử đơn vị** 9](#_Toc89260601)

[**3.** **KẾ HOẠCH KIỂM THỬ** 10](#_Toc89260602)

[**4.** **TRIỂN KHAI KIỂM THỬ HỘP TRẮNG CHO ỨNG DỤNG** 11](#_Toc89260603)

[**4.1.** **Kiểm thử chức năng Login (API)** 11](#_Toc89260604)

[**4.1.1.** **Mã nguồn** 11](#_Toc89260605)

[**4.1.2.** **Biểu đồ luồng điều khiển (Control-flow graph)** 13](#_Toc89260606)

[**4.1.3.** **Độ phức tạp Cyclomatic và đường thực thi tuyến tính độc lập** 13](#_Toc89260607)

[**4.1.4.** **Thiết kế test cases cho mỗi đường** 13](#_Toc89260608)

[**4.1.5.** **Biểu diễn luồng dữ liệu cho tất cả các biến** 16](#_Toc89260609)

[**4.1.6.** **Biểu diễn luồng dữ liệu để kiểm thử từng biến** 16](#_Toc89260610)

[**4.2.** **Kiểm thử chức năng Tạo mới câu hỏi (API)** 25](#_Toc89260611)

[**4.2.1.** **Mã nguồn** 25](#_Toc89260612)

[**4.2.2.** **Biểu đồ luồng điều khiển (Control-flow graph)** 26](#_Toc89260613)

[**4.2.3.** **Độ phức tạp Cyclomatic và đường thực thi tuyến tính độc lập** 27](#_Toc89260614)

[**4.2.4.** **Thiết kế test cases cho mỗi đường** 27](#_Toc89260615)

[**4.2.5.** **Biểu diễn luồng dữ liệu cho tất cả các biến** 28](#_Toc89260616)

[**4.2.6.** **Biểu diễn luồng dữ liệu để kiểm thử từng biến** 29](#_Toc89260617)

[**4.3.** **Kiểm thử chức năng** 33](#_Toc89260618)

[**4.3.1.** **Mã nguồn** 33](#_Toc89260619)

[**4.3.2.** **Biểu đồ luồng điều khiển (Control-flow graph)** 33](#_Toc89260620)

[**4.3.3.** **Độ phức tạp Cyclomatic và đường thực thi tuyến tính độc lập** 33](#_Toc89260621)

[**4.3.4.** **Thiết kế test cases cho mỗi đường** 33](#_Toc89260622)

[**4.3.5.** **Biểu diễn luồng dữ liệu cho tất cả các biến** 33](#_Toc89260623)

[**4.3.6.** **Biểu diễn luồng dữ liệu để kiểm thử từng biến** 33](#_Toc89260624)

[**4.4.** **Kiểm thử chức năng** 33](#_Toc89260625)

[**4.4.1.** **Mã nguồn** 33](#_Toc89260626)

[**4.4.2.** **Biểu đồ luồng điều khiển (Control-flow graph)** 33](#_Toc89260627)

[**4.4.3.** **Độ phức tạp Cyclomatic và đường thực thi tuyến tính độc lập** 33](#_Toc89260628)

[**4.4.4.** **Thiết kế test cases cho mỗi đường** 33](#_Toc89260629)

[**4.4.5.** **Biểu diễn luồng dữ liệu cho tất cả các biến** 33](#_Toc89260630)

[**4.4.6.** **Biểu diễn luồng dữ liệu để kiểm thử từng biến** 33](#_Toc89260631)

[**5.** **TRIỂN KHAI KIỂM THỬ HỘP ĐEN CHO ỨNG DỤNG** 34](#_Toc89260632)

[TÀI LIỆU THAM KHẢO 35](#_Toc89260633)

[PHỤ LỤC 36](#_Toc89260634)

[PHÂN CÔNG CÔNG VIỆC 37](#_Toc89260635)

# **LỜI GIỚI THIỆU**

# **NỘI DUNG**

## **ĐẶC TẢ, MÔ HÌNH HÓA YÊU CẦU VÀ GIỚI THIỆU CÁC CHỨC NĂNG CỦA ỨNG DỤNG**

### ***Đặc tả***

#### **Yêu cầu chức năng**

Xây dựng một hệ thống website cho phép:

* Tạo các cuộc thi. Mỗi cuộc thi có thông thi chi tiết mô tả về cuộc thi, thời gian diễn ra, URL truy cập.
* Mỗi cuộc thi bao gồm nhiều bài thi. Thông tin chi tiết về bài thi bao gồm mô tả bài thi, tên bài thi, thời gian bắt đầu và kết thúc, URL truy cập.
* Mỗi bài thi gồm nhiều câu hỏi để cho thí sinh trả lời. Các câu hỏi có thể có thêm hình ảnh minh họa cho trực quan, sinh động.
* Các thi sinh có thể truy cập vào cuộc thi để tham gia và làm cái bài thi. Khi làm bài, hệ thống sẽ đếm ngược thời gian cho thí sinh.
* Người tạo đề thi có thể chia sẻ các link cuộc thi, link bài thi để thí sinh làm bài. Ngoài ra cũng có thể xem thống kê chi tiết về bài thi, bao gồm số lượng thí sinh tham gia, chi tiết về các bài thi.

#### **Yêu cầu phi chức năng**

* Giao diện trực quan, sinh động, dễ thao tác.
* Tốc độ xử lý và phản hồi của hệ thống nhanh chóng.
* Bảo mật cơ sở dữ liệu, không để lọt lộ thông tin, cũng như kết quả làm bài.

### ***Mô hình hóa yêu cầu***

#### **Sơ đồ phân rã chức năng**

|  |
| --- |
|  |

#### **Lược đồ use case tổng quan**

*Client side:*

|  |
| --- |
|  |

* *Admin side:*

|  |
| --- |
|  |

#### **Lược đồ use case chi tiết**

|  |
| --- |
| **Diagram  Description automatically generated** |
|  |
|  |
|  |
|  |
| **Diagram  Description automatically generated** |
| **Diagram  Description automatically generated** |
| **Diagram  Description automatically generated** |
|  |
|  |
| Diagram  Description automatically generated |
| Diagram  Description automatically generated |
| Diagram  Description automatically generated |
|  |

#### **Scenarios**

|  |  |
| --- | --- |
| **Name** | **Register – Đăng ký tài khoản** |
| Brief description | Khách (Guest) truy cập tạo mới tài khoản. |
| Actor(s) | Guest |
| Pre-conditions | Không |
| Post-conditions | - Nếu đăng ký thành công: Người dùng được tạo mới tài khoản, thông tin cá nhân được lưu vào CSDL.  - Nếu đăng ký không thành công: Thông báo không tạo được tài khoản, buộc người dùng nhập lại thông tin cho chính xác. |
| Flow of events |  |
| Basic flow | Use case bắt đầu khi khách truy cập vào trang tạo tài khoản.  1. Hệ thống hiển thị biểu mẫu cho khách nhập chọn loại người dùng: *Giáo viên* hay *Học sinh/ Sinh viên*.  2. Khách chọn loại người dùng muốn đăng ký.  3. Hệ thống lưu lại thông tin và hiển thị biểu mẫu tương ứng với loại người dùng khách chọn.  4. Khách điền vào các thông tin mà hệ thống yêu cầu rồi nhấn ĐĂNG KÝ.  5. Hệ thống xác thực thông tin theo quy định.  6. Hệ thống tạo mới tài khoản và lưu thông tin vào CSDL.  7. Hệ thống thông báo tạo tài khoản thành công và chuyển đến trang cá nhân của người dùng. |
| Alternative flow | Nếu người dùng nhập thiếu thông tin, trùng email hoặc mật khẩu không đúng quy định, khi đó hệ thống sẽ:  1. Hệ thống mô tả lý do không thể tạo mới tài khoản.  2. Hệ thống hiển thị lại biểu mẫu cho người dùng chỉnh sửa thông tin đăng ký.  3. Người dùng nhập lại thông tin được yêu cầu, Basic Flow khi đó sẽ tiếp tục tại bước 3. |
| Extension point | Không có |

|  |  |
| --- | --- |
| **Name** | **Login – Đăng nhập.** |
| Brief description | Người dùng đăng nhập vào hệ thống |
| Actor(s) | Creator, Examinee, Admin, Super Admin |
| Pre-conditions | Actors đã có tài khoản trong hệ thống |
| Post-conditions | - Sau khi đăng nhập: người dùng được xác thực và vào trang được chỉ định tùy theo từng chức vụ.  - Đăng nhập thất bại: thông báo lỗi đăng nhập và yêu cầu đăng nhập lại. |
| Flow of events |  |
| Basic flow  (Thành công) | Use case được kích hoạt khi người dùng cần đăng nhập vào hệ thống hoặc xác thực danh tính để sử dụng các chức năng của hệ thống:  1. Người nhập username và password vào các ô input hoặc chọn Đăng nhập bằng tài khoản Google.  2. Hệ thống kiểm tra input và xác thực  3. Hệ thống lưu lại session đăng nhập.  4. Hệ thống thông báo xác thực thành công. |
| Alternative flow  (Thất bại) | Khi người dùng sai username hoặc password:  1. Hệ thống thông báo lỗi sai  2. Người dùng nhập lại thông tin sai  Hệ thống xác thực lại, quay lại các bước ở basic flow trên.  Khi hệ thống không phát hiện tài khoản Google tương thích đã được đăng nhập:  1. Hệ thống thông báo không tìm thấy tài khoản Google và |
| Extension point | Người dùng đăng nhập vào hệ thống |

|  |  |
| --- | --- |
| **Name** | **Logout – Đăng xuất** |
| Brief description | Đăng xuất tài khoản người dùng khỏi hệ thống. |
| Actor(s) | Examinee, Creator, Admin, Super Admin |
| Pre-conditions | Actors đã đăng nhập thành công vào hệ thống. |
| Post-conditions | Tài khoản được đăng xuất thành công ra khỏi hệ thống. |
| Flow of events |  |
| Basic flow  (Thành công) | 1. Người dùng bấm vào Đăng xuất  2. Hệ thống hiển thị hộp thoại xác nhận việc đăng xuất: Bạn có muốn đăng xuất khỏi hệ thống không? (Có/Không)  3. Người dùng chọn Có.  4. Tài khoản người dùng được đăng xuất khỏi hệ thống, đồng thời hiện lên thông báo “Bạn đã đăng xuất thành công khỏi hệ thống.” |
| Alternative flow  (Thất bại) | Khi người dùng chọn “Không” ở bước 3 (basic flow), hộp thoại xác nhận sẽ được đóng lại. |
| Extension point |  |

|  |  |
| --- | --- |
| **Name** | **Password retrieval – Lấy lại mật khẩu** |
| Brief description | Cấp lại mật khẩu cho người dùng qua email khi người dùng quên mật khẩu. |
| Actor(s) | Examinee, Creator, Admin, Super Admin |
| Pre-conditions | Actors đã có tài khoản trong hệ thống. |
| Post-conditions | Tài khoản của người dùng được cập nhật mật khẩu mới. |
| Flow of events |  |
| Basic flow  (Thành công) | 1. Người dùng bấm vào nút Lấy lại mật khẩu trên trang Đăng nhập.  2. Hệ thống hiển thị hộp thoại yêu cầu người dùng nhập email của tài khoản cần lấy lại mật khẩu.  3. Người dùng nhập email và bấm Gửi.  4. Hệ thống gửi 01 mật khẩu tạm thời về email của người dùng, đồng thời hiện form yêu cầu người dùng nhập vào: Mật khẩu tạm đã nhận (được gửi về email), mật khẩu mới, xác nhận mật khẩu mới.  5. Người dùng nhập vào Mã code (được gửi về email), mật khẩu mới, xác nhận mật khẩu mới rồi bấm Xác nhận.  6. Hệ thống thông báo đã cập nhật mật khẩu thành công, chuyển hướng đến trang đăng nhập. |
| Alternative flow  (Thất bại) | *Khi người dùng nhập email không tồn tại trong hệ thống:*  1. Thông báo cho người dùng rằng không có tài khoản tương ứng với email đã nhập.  2. Hiện lại form để người dùng nhập lại email.  *Khi người dùng nhập sai mật khẩu tạm:*  1. Thông báo cho người dùng rằng mật khẩu tạm đã nhập không hợp lệ, yêu cầu người dùng kiểm tra lại email. |
| Extension point |  |

|  |  |
| --- | --- |
| **Name** | **Create a Subject – Tạo mới chủ đề** |
| Brief description | Creator tạo mới chủ đề của bài thi. |
| Actor(s) | Creator |
| Pre-conditions | Actors đã đăng nhập vào hệ thống. |
| Post-conditions | Chủ đề mới được lưu thành công vào CSDL. |
| Flow of events |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng bấm chọn Quản lý chủ đề, hệ thống hiển thị các chủ đề đã có.  1. Creator chọn Tạo mới.  2. Hệ thống hiển thị biểu mẫu điền thông tin về chủ đề mới.  3. Creator điền thông tin và bấm Create.  4. Chủ đề mới được lưu vào CSDL và hệ thống thông báo chủ đề được tạo thành công. |
| Alternative flow  (Thất bại) | Khi người dùng nhập thiếu các thông tin cần thiết để tạo chủ đề:  1. Thông báo người dùng bổ sung thông tin cho chủ đề.  2. Quay lại bước 2 của Basic flow. |
| Extension point |  |

|  |  |
| --- | --- |
| **Name** | **Modify Subject – Chỉnh sửa chủ đề** |
| Brief description | Creator chỉnh sửa chủ đề của bài thi. |
| Actor(s) | Creator |
| Pre-conditions | Actors đã đăng nhập vào hệ thống. |
| Post-conditions | Thông tin chi tiết của chủ đề được cập nhật thành công vào CSDL. |
| Flow of events |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng bấm chọn Quản lý chủ đề, hệ thống hiển thị các chủ đề đã có.  1. Creator nhấn vào chủ đề tương ứng, bấm chọn nút Chỉnh sửa.  2. Hệ thống hiển thị biểu mẫu chỉnh sửa thông tin.  3. Creator nhập lại thông tin cần chỉnh sửa và bấm Lưu.  4. Hệ thống thông báo chủ đề được cập nhật. |
| Alternative flow  (Thất bại) | Khi người dùng bấm nút Hủy bỏ:  1. Hiện hộp thông thông báo xác nhận: “Thông tin chỉnh sửa chưa được lưu. Bạn có muốn tiếp tục?” (Có/Không).  2. Người dùng bấm “Có”: các thay đổi sẽ không được lưu, quay lại trang xem danh sách chủ đề. Người dùng bấm “Không”: đóng hộp thoại xác nhận, tiếp tục chỉnh sửa. |
| Extension point |  |

|  |  |
| --- | --- |
| **Name** | **Create a Subject – Tạo mới chủ đề** |
| Brief description | Creator xóa chủ đề của bài thi. |
| Actor(s) | Creator |
| Pre-conditions | Actors đã đăng nhập vào hệ thống. |
| Post-conditions | Hệ thống thông báo xóa chủ đề thành công, đồng thời chủ đề tương ứng bị xóa khỏi CSDL. |
| Flow of events |  |
| Basic flow  (Thành công) | Use case bắt đầu khi người dùng bấm chọn Quản lý chủ đề, hệ thống hiển thị các chủ đề đã có.  1. Creator nhấn vào chủ đề tương ứng, bấm chọn nút Delete.  2. Hệ thống hiển thị hộp thoại xác nhận xóa.  3. Creator bấm OK để xóa.  4. Hệ thống thông báo chủ đề được xóa thành công, đồng thời chủ đề tương ứng trong CSDL cũng bị xóa. |
| Alternative flow  (Thất bại) | Khi người dùng bấm Hủy: quay lại trang xem danh sách chủ đề. |
| Extension point |  |

|  |  |
| --- | --- |
| **Name** | **Create contest – Tạo cuộc thi.** |
| Brief description | Creator tạo mới cuộc thi. |
| Actor(s) | Creator |
| Pre-conditions | Actors đã đăng nhập vào hệ thống. |
| Post-conditions | Hệ thống thông báo tạo cuộc thi thành công, đồng thời thông tin về cuộc thi được lưu vào CSDL. |
| Flow of events |  |
| Basic flow  (Thành công) | Use case bắt đầu khi Creator đang ở trang Quản lý cuộc thi:  1. Creator bấm chọn nút Tạo cuộc thi.  2. Hệ thống hiện biểu mẫu về các thông tin cần thiết của một cuộc thi.  3. Creator nhập thông tin chi tiết về cuộc thi vào biểu mẫu và bấm Hoàn thành.  4. Thông tin cuộc thi được lưu vào CSDL, đồng thời hệ thống thông báo tạo cuộc thi mới thành công. |
| Alternative flow  (Thất bại) | Khi người dùng bấm Hủy bỏ:  1. Hệ thống hiện hộp thoại xác nhận việc hủy khởi tạo cuộc thi.  2. Người dùng xác nhận hủy.  3. Hệ thống chuyển hướng về trang Quản lý cuộc thi. |
| Extension point |  |

|  |  |
| --- | --- |
| **Name** | **Update contest – Cập nhật cuộc thi.** |
| Brief description | Creator cập nhật thông tin cuộc thi. |
| Actor(s) | Creator |
| Pre-conditions | Actors đã đăng nhập vào hệ thống. |
| Post-conditions | Hệ thống thông báo tạo đã cập nhật cuộc thi thành công, đồng thời thông tin về cuộc thi được lưu vào CSDL. |
| Flow of events |  |
| Basic flow  (Thành công) | Use case bắt đầu khi Creator đang ở trang Quản lý cuộc thi:  1. Creator bấm chọn nút Update.  2. Hệ thống hiện biểu mẫu về các thông tin cần thiết của một cuộc thi.  3. Creator nhập thông tin chi tiết về cuộc thi vào biểu mẫu và bấm Lưu.  4. Thông tin cuộc thi được lưu vào CSDL, đồng thời hệ thống thông báo thông tin cuộc thi đã được cập nhật thành công. |
| Alternative flow  (Thất bại) | Khi người dùng bấm Hủy bỏ:  1. Hệ thống hiện hộp thoại xác nhận việc hủy cập nhật cuộc thi.  2. Người dùng xác nhận hủy.  3. Hệ thống chuyển hướng về trang Quản lý cuộc thi. |
| Extension point |  |

|  |  |
| --- | --- |
| **Name** | **Publish contest – Công bố cuộc thi.** |
| Brief description | Creator công bố cuộc thi. |
| Actor(s) | Creator |
| Pre-conditions | Actors đã đăng nhập vào hệ thống. |
| Post-conditions | Hệ thống thông báo tạo đã công bố cuộc thi thành công. |
| Flow of events |  |
| Basic flow  (Thành công) | Use case bắt đầu khi Creator đang ở trang Quản lý cuộc thi:  1. Creator bấm chọn nút Publish.  2. Hệ thống hiện biểu mẫu xác nhận việc công bố cuộc thi.  3. Creator chọn Đồng ý.  4. Thông tin cuộc thi được cập nhật vào CSDL, đồng thời hệ thống thông báo cuộc thi đã được công bố thành công và chuyển hướng đến trang chính cuộc thi. |
| Alternative flow  (Thất bại) | Khi người dùng bấm Hủy bỏ: Hệ thống thông báo cho Creator về việc hủy bỏ công bố cuộc thi, đồng thời chuyển hướng về trang Quản lý cuộc thi. |
| Extension point |  |

|  |  |
| --- | --- |
| **Name** | **Archive contest – Lưu trữ cuộc thi.** |
| Brief description | Creator lưu trữ cuộc thi. |
| Actor(s) | Creator |
| Pre-conditions | Actors đã đăng nhập vào hệ thống. |
| Post-conditions | Hệ thống thông báo tạo đã lưu trữ cuộc thi thành công. |
| Flow of events |  |
| Basic flow  (Thành công) | Use case bắt đầu khi Creator đang ở trang Quản lý cuộc thi:  1. Creator bấm chọn nút Lưu trữ.  2. Hệ thống hiện biểu mẫu xác nhận việc lưu trữ cuộc thi.  3. Creator chọn đồng ý.  4. Thông tin cuộc thi được cập nhật vào CSDL, đồng thời hệ thống thông báo cuộc thi đã được lưu trữ thành công. |
| Alternative flow  (Thất bại) | Khi người dùng bấm Hủy bỏ: Hệ thống thông báo cho Creator về việc hủy bỏ lưu trữ cuộc thi, đồng thời chuyển hướng về trang Quản lý cuộc thi. |
| Extension point |  |

### ***Các chức năng của ứng dụng***

## **CƠ SỞ LÝ THUYẾT VỀ KIỂM THỬ**

### ***Kiểm thử đơn vị***

## **KẾ HOẠCH KIỂM THỬ**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Week** | **Duration** | **Tasks** | **Student in charge** | **Expected results / products** | **Date of completion** | **Completion Rate** |
| 1 | 1/11 – 7/11 | 1. Tìm hiểu về các phương pháp và kỹ thuật kiểm thử cơ bản.  2. Đặc tả và mô hình hóa các yêu cầu hệ thống và mô tả các chức năng của hệ thống. | 1. Nguyễn Thị Minh Hoàng + Lê Thị Ngọc Yến  2. Trần Ngọc Minh Thiện + Nguyễn Huỳnh Minh Tiến | 1. Hiểu các kỹ thuật kiểm thử cơ bản: unit testing, integration test, system test, acceptance test…  2. Các use cases, đặc tả use cases, biểu đồ phân rã chức năng của hệ thống… |  |  |
| 2 | 8/11 – 14/11 | 1. Lập kế hoạch kiểm thử dự án.  2. Lựa chọn phương pháp kiểm thử thích hợp để áp dụng kiểm thử cho hệ thống. | 1. Nguyễn Thị Minh Hoàng + Lê Thị Ngọc Yến  2. Trần Ngọc Minh Thiện + Nguyễn Huỳnh Minh Tiến | 1. Hoàn thành kế hoạch kiểm thử.  2. Xác định được phương pháp kiểm thử cho hệ thống và hoàn thành cơ sở lý thuyết cho đề tài. |  |  |
| 3 | 15/11 – 21/11 | Tiến hành kiểm thử: chọn một số đoạn code / hàm / phương thức / module… để kiểm thử. | - Nguyễn Thị Minh Hoàng  - Lê Thị Ngọc Yến  - Trần Ngọc Minh Thiện  - Nguyễn Huỳnh Minh Tiến | - Tạo đồ thị dòng điều khiển. Tính toán độ phức tạp Cyclomatic để xác định các đường thực thi tuyến tính độc lập. Thiết kế các trường hợp kiểm thử cho mỗi đường dẫn.  - Tạo biểu đồ luồng dữ liệu chung. Thực hiện kiểm tra vòng đời cho mỗi biến trong đơn vị cho mỗi đơn vị code đã chọn.  - Sửa các lỗi trong đồ thị nếu có |  |  |
| 4 | 22/11 – 28/11 | 1. Dựa vào 08 kỹ thuật kiểm thử hộp đen để thiết kế ít nhất 50 bộ test case cho hệ thống  2. Viết bug reports cho các test case không Pass ở các bộ test case phía trên | - Nguyễn Thị Minh Hoàng  - Lê Thị Ngọc Yến  - Trần Ngọc Minh Thiện  - Nguyễn Huỳnh Minh Tiến | 1. Danh sách các bộ kiểm thử sử dụng kỹ thuật hộp đen.  2. Kết quả kiểm thử và Bug reports cho các test cases đã lên danh sách. |  |  |
| 5 | 29/11 – 5/12 | Tiến hành sửa lỗi. | Trần Ngọc Minh Thiện + Nguyễn Huỳnh Minh Tiến | Sửa các lỗi đã được báo cáo trong Bug reports |  |  |
| 6 | 6/12 – 12/12 | Hoàn thành báo cáo, thực hiện format báo cáo theo yêu cầu chung | Nguyễn Thị Minh Hoàng + Lê Thị Ngọc Yến | Hoàn thành báo cáo |  |  |

## **TRIỂN KHAI KIỂM THỬ HỘP TRẮNG CHO ỨNG DỤNG**

### ***Kiểm thử chức năng Login (API)***

#### **Mã nguồn**

|  |
| --- |
| 1. router.post("/login", async (req, res) => { 2. const { username, password } = req.body; **[1]** 3. //simple validation 4. if (!username **[2]** || !password **[3]**) { 5. return res 6. .status(400) 7. .json( 8. { 9. success: false, 10. message: "Missing username and/or password" 11. } 12. ); **[4]** 13. } 14. try { 15. //check for existing username 16. const account = await Account.findOne({ username }); **[5]** 17. if (!account) **[6]** 18. return res 19. .status(400) 20. .json( 21. { 22. success: false, 23. message: "Incorrect username or password" 24. } 25. ); **[7]** 26. // Username found 27. const passwordValid 28. = await argon2.verify(account.password, password); **[8]** 29. if (!passwordValid) **[9]** 30. return res 31. .status(400) 32. .json( 33. { 34. success: false, 35. message: "Incorrect username or password" 36. } 37. ); **[10]** 38. //All good 39. const userInfo = await User.findOne({ account: account.\_id }); **[11]** 40. //return token 41. const accessToken = jwt.sign( 42. { 43. verifyAccount: 44. { 45. id: account.\_id, 46. isHidden: account.isHidden, 47. username: account.username, 48. role: account.role, 49. }, 50. }, 51. process.env.ACCESS\_TOKEN\_SECRET 52. ); **[12]** 53. res.json( 54. { 55. success: true, 56. message: "User logged successfully", 57. accessToken: accessToken, 58. user: userInfo, 59. }); **[13]** 60. } 61. catch (error) { 62. console.log(error); **[14]** 63. res.status(500).json( 64. { 65. success: false, 66. message: "Internal server error" 67. } 68. ); **[15]** 69. } 70. }); |

#### **Biểu đồ luồng điều khiển (Control-flow graph)**

|  |
| --- |
|  |

#### **Độ phức tạp Cyclomatic và đường thực thi tuyến tính độc lập**

V= 24 - 17+2 = 9

#### **Thiết kế test cases cho mỗi đường**

|  |  |  |
| --- | --- | --- |
| **Path** | **Data** | **Expected result** |
| 1-2-3-4 | req.body = {    username: "",    password: "1245"  } | {    success: false,    message: "Missing username and/or password"  } |
| 1-2-3-5-6-7 | req.body =  {    username: 'admin',    password: '12345'  } | {    success: false,    message: "Incorrect username or password"  } |
| 1-2-3-5-14 | req.body =  {    username: '',    password: '12345'  } |  |
| 1-2-3-5-6-8-9-10 | req.body =  {    username: 'minhhoang1',    password: '123'  } | {    success: false,    message: "Incorrect username or password"  } |
| 1-2-3-5-6-8-9-11-12-13 | req.body =  {    username: 'minhhoang1',    password: '12345'  } | {    "success": true,    "message": "User logged successfully",    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6  IkpXVCJ9.eyJ2ZXJpZnlBY2NvdW5  0Ijp7ImlkIjoiNjE4ZGNlYThhNjE  xZjQzNDAyOTYzMjhjIiwidXNlcm5  hbWUiOiJtaW5oaG9hbmcxIiwicm9  sZSI6ImFkbWluIn0sImlhdCI6MTY  zODMyNjAzNX0.7r4ILoIHjxlmWn5  D2el4yDFzy6X8PMMpl - 7PZVzD\_24",    "user": {      "fullname": "Nguyen Anh Tin",      "address": null,      "dod": null,      "school": null,      "phone": "124567855",      "account": "618dcea8a611f4340296328c",      "avatar": null,      "isHidden": false,      "createdAt": "2021-11-12T09:17:12.000Z",      "updatedAt": "2021-11-12T09:17:12.000Z",      "id": "618dcea8a611f4340296328e"    }  } |
| 1-2-3-5-6-8-14-15 | req.body =  {    username: 'minhhoang1',    password: '12345'  }  // xảy ra lỗi trong quá trình giải mã argon2 | {    success: false,    message: "Internal server error"  } |
| 1-2-3-5-6-8-9-11-14-15 | req.body =  {    username: 'minhhoang1',    password: '12345'  }  // xảy ra expeption trong quá trình truy suất mongoDB | {    success: false,    message: "Internal server error"  } |
| 1-2-3-5-6-8-9-11-12-14-15 | req.body =  {    username: 'minhhoang1',    password: '12345'  }  // xảy ra lỗi trong quá trình tạo accessToken bằng jwt | {    success: false,    message: "Internal server error"  } |

#### **Biểu diễn luồng dữ liệu cho tất cả các biến**

|  |
| --- |
|  |

#### **Biểu diễn luồng dữ liệu để kiểm thử từng biến**

* *Kịch bản biến req đều là ~duk:*

|  |
| --- |
|  |

Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

* *Kịch bản biến res đều là ~duk*

|  |
| --- |
|  |

Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

* *Kịch bản biến username đều là ~duk và duuk*

|  |
| --- |
|  |

Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

* *Kịch bản biến password đều là ~dk, ~duk, ~duuk*

|  |
| --- |
|  |

Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

* *Kịch bản biến account đều là ~d k, ~duk, ~duuk, ~duuuk, ~duuuuk*

|  |
| --- |
| A picture containing necklet, scale, accessory  Description automatically generated |

Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

* *Kịch bản biến passwordValid đều là ~dk, ~duk*

|  |
| --- |
| A picture containing necklet, scale, accessory, scissors  Description automatically generated |

Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

* *Kịch bản biến userInfo đều là ~dk, ~duk*

|  |
| --- |
|  |

Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

* *Kịch bản biến error đều là ~duk*

|  |
| --- |
|  |

Các kịch bản không chứa cặp đôi nào hoạt động bất thường.

### ***Kiểm thử chức năng Tạo mới câu hỏi (API)***

#### **Mã nguồn**

|  |
| --- |
| 1. router.post("/new/:testId", verifyToken, async (req, res) => { 2. try { 3. //Check permission 4. if ( 5. !( 6. req.body.verifyAccount.role === ROLES.ADMIN **[1]** || 7. req.body.verifyAccount.role === ROLES.CREATOR **[2]** 8. ) 9. ) { 10. return res 11. .status 12. .json({ success: false, message: "Permission denied" }); **[3]** 13. } 14. if (!req.body) **[4]** 15. res.status(400).json( 16. { success: false, message: "Body request not found" } 17. ); **[5]** 18. //Create new question 19. let question = new Question({ 20. order: req.body.order, 21. content: req.body.content, 22. type: req.body.type, 23. answers: req.body.answers, 24. correctAnswers: req.body.correctAnswers, 25. embededMedia: req.body.embededMedia, 26. }); **[6]** 27. //Send to Database 28. question = await question.save();**[7]** 29. // Updating for Test Collection : Question list 30. let test = await Test.findById(req.params.testId); **[8]** 31. console.log(test); 32. if (test) **[9]** 33. { 34. test.questions.push(question.id.toString());**[10]** 35. test = await Test.findOneAndUpdate( 36. { \_id: test.id }, 37. test, 38. { new: true } 39. ); **[11]** 40. } 41. res.json( 42. { 43. success: true, 44. message: "Question created successfully", 45. question: question, 46. } 47. ); **[12]** 48. } 49. catch (error) { 50. console.log(error); 51. res.status(500).json( 52. { success: false, message: "Internal server error" } 53. ); **[13]** 54. } 55. }); |

#### **Biểu đồ luồng điều khiển (Control-flow graph)**

|  |
| --- |
|  |

#### **Độ phức tạp Cyclomatic và đường thực thi tuyến tính độc lập**

Có 4 nút nhị phân => C = 7 + 1 = 8

#### **Thiết kế test cases cho mỗi đường**

|  |  |  |
| --- | --- | --- |
| **Path** | **Data** | **Expected result** |
| 1-> 2 -> 3 -> 13 | req.body.verifyAccount.role = ROLES.USER | {    success: false,    message: "Permission denied"  } |
| 1 -> 2 -> 4 -> 5 | req.body.verifyAccount.role = ROLES.ADMIN  req.body = null | {    success: false,    message: "Body request not found"  } |
| 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 9 -> 12 | req.body.verifyAccount.role = ROLES.ADMIN  req.body != null  test = null | {    success: true,    message: "Question created successfully",    question: question  } |
| 1 -> 2 -> 4 -> 6 -> 7 -> 13 | req.body.verifyAccount.role = ROLES.ADMIN  req.body != null | Throw exception (do bị trùng các field đặt ràng buộc unique) |
| 1 -> 2 -> 4 -> 6 -> 7 --> 8 -> 13 | req.body.verifyAccount.role = ROLES.ADMIN  req.body != null | Throw exception (do lỗi MongoDB) |
| 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 9 -> 10 -> 11 ->12 | req.body.verifyAccount.role = ROLES.ADMIN  req.body != null  test = null | {    success: true,    message: "Question created successfully",    question: question  } |
| 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 9 -> 10 -> 13 | req.body.verifyAccount.role = ROLES.ADMIN  req.body != null  test != null  question.id = null | Throw exception |
| 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 9 -> 10 -> 11 ->13 | req.body.verifyAccount.role = ROLES.ADMIN  req.body != null  test != null  question.id != null | Throw exception (do lỗi MongoDB) |

#### **Biểu diễn luồng dữ liệu cho tất cả các biến**

|  |
| --- |
|  |

#### **Biểu diễn luồng dữ liệu để kiểm thử từng biến**

* *Kiểm thử biến req*

|  |
| --- |
|  |

Các dòng duk và duuk đều không chứa cặp đôi nào hoạt động bất thường.

* *Kiểm thử biến res*

|  |
| --- |
|  |

Các dòng duuk đều không chứa cặp đôi nào hoạt động bất thường.

* *Kiểm thử biến question*

|  |
| --- |
|  |

*Các dòng duk đều không chứa cặp đôi nào bất thường.*

* *Kiểm thử biến test*

|  |
| --- |
|  |

Các dòng dk, duk, duuk đều không chứa cặp đôi bất thường.

### ***Kiểm thử chức năng***

#### **Mã nguồn**

|  |
| --- |
| 1. const onTestAction = (action) => { 2. if (currentAction === 'CONFIRM\_DELETE\_TEST') **[1]** 3. { 4. if (selectedTest **[2]** && 5. action === MODAL\_ACTION\_CONFIRM **[3]**) 6. { 7. const newContest = { ...contest } **[4]** 8. const index = 9. newContest.tests.findIndex(c => c.id === selectedTest.id) **[5]** 10. newContest.tests.splice(index, 1) **[6]** 11. updateContest(newContest) **[7]** 12. } 13. } 14. else if (currentAction === 'CONFIRM\_CREATE\_TEST') **[8]** 15. { 16. if (action === MODAL\_ACTION\_CONFIRM) **[9]** 17. { 18. const newContest = { ...contest } **[10]** 19. newContest.tests.push( 20. { 21. ...emptyTest, 22. id: 'test-' + (contest.tests.length + 1) 23. }) **[11]** 24. updateContest(newContest) **[12]** 25. const index = newContest.tests.length – 1 **[13]** 26. setSelectedTest(newContest.tests[index]) **[14]** 27. setIsShowTest(true) **[15]** 28. } 29. } 30. setIsShowConfirmModal(false) **[16]** 31. } |

#### **Biểu đồ luồng điều khiển (Control-flow graph)**

|  |
| --- |
|  |

#### **Độ phức tạp Cyclomatic và đường thực thi tuyến tính độc lập**

Có 5 nút nhị phân => V = 5 + 1 = 6

#### **Thiết kế test cases cho mỗi đường**

|  |  |  |
| --- | --- | --- |
| **Path** | **Data** | **Expected result** |
| 1-2-16 | currentAction = 'CONFIRM\_DELETE\_TEST'  selectedTest = null  contest != null | - isShowConfirmModal = true  - Hộp thoại Confirm được ẩn đi |
| 1-2-3-16 | currentAction = 'CONFIRM\_DELETE\_TEST'  action = 'MODAL\_ACTION\_CLOSE'  selectedTest != null  contest != null | - isShowConfirmModal = true  - Hộp thoại Confirm được ẩn đi |
| 1-2-3-4-5-6-7-16 | currentAction = 'CONFIRM\_DELETE\_TEST'  selectedTest != null  action = 'MODAL\_ACTION\_CONFIRM'  contest != null | - isShowConfirmModal = true  - Giá trị của contest được cập nhật  - Hộp thoại Confirm được ẩn đi |
| 1-8-16 | currentAction = 'CONFIRM'  contest != null | - isShowConfirmModal = true  - Hộp thoại Confirm được |
| 1-8-9-16 | currentAction = 'CONFIRM\_CREATE\_TEST'  action = 'MODAL\_ACTION\_CLOSE' | - isShowConfirmModal = true  - Hộp thoại Confirm được ẩn đi |
| 1-8-9-10-11-12-13-14-15-16 | currentAction = 'CONFIRM\_CREATE\_TEST'  action = 'MODAL\_ACTION\_CONFIRM'  emptyTest != null | - isShowConfirmModal = true  - Hộp thoại Confirm được ẩn đi  - Một bài test được tạo mới và thêm vào cuối danh sách của contest.tests |

#### **Biểu diễn luồng dữ liệu cho tất cả các biến**

|  |
| --- |
|  |

#### **Biểu diễn luồng dữ liệu để kiểm thử từng biến**

* *Kiểm thử luồng dữ liệu biến* currentAction

|  |
| --- |
|  |

* Kịch bản 1: ~du
* Kịch bản 2: ~duu

Các kịch bản đều không chứa các bất thường.

* *Kiểm thử luồng dữ liệu biến* selectedTest

|  |
| --- |
|  |

* Kịch bản 1: ~d
* Kịch bản 2: ~du
* Kịch bản 3: ~duu
* Kịch bản 4: ~dd

Các kịch bản trên đều không chứa các bất thường.

* Kiểm thử luồng dữ liệu biến action

|  |
| --- |
|  |

* Kịch bản 1: ~d
* Kịch bản 2: ~du

Các kịch bản trên đều không chứa các bất thường

* *Kiểm thử luồng dữ liệu biến* contest

|  |
| --- |
|  |

* Kịch bản 1: ~d
* Kịch bản 2: ~dud

Các kịch bản trên đều không chứa các bất thường.

* *Kiểm thử luồng dữ liệu biến* newContest

|  |
| --- |
|  |

* Kịch bản 1: ~dudu
* Kịch bản 2: ~dduuu

Các kịch bản trên đều không chứa các bất thường.

* *Kiểm thử luồng dữ liệu biến* index

|  |
| --- |
|  |

* Kịch bản 1: ~du
* Kịch bản 2: ~du

Các kịch bản trên đều không chứa các bất thường.

* *Kiểm thử luồng dữ liệu biến* emptyTest

|  |
| --- |
|  |

* Kịch bản 1: ~d
* Kịch bản 2: ~du

Các kịch bản trên đều không có bất thường.

* *Kiểm thử luồng dữ liệu biến* isShowTest

|  |
| --- |
|  |

* Kịch bản 1: ~d
* Kịch bản 2: ~dd

Các kịch bản trên đều không chứa các bất thường.

* *Kiểm thử luồng dữ liệu biến* isShowConfirmModal

|  |
| --- |
|  |

* Kịch bản: ~dd => Không chứa bất thường

### ***Kiểm thử chức năng***

#### **Mã nguồn**

|  |
| --- |
| x |

#### **Biểu đồ luồng điều khiển (Control-flow graph)**

#### **Độ phức tạp Cyclomatic và đường thực thi tuyến tính độc lập**

#### **Thiết kế test cases cho mỗi đường**

#### **Biểu diễn luồng dữ liệu cho tất cả các biến**

#### **Biểu diễn luồng dữ liệu để kiểm thử từng biến**

## **TRIỂN KHAI KIỂM THỬ HỘP ĐEN CHO ỨNG DỤNG**

Dựa vào 08 kỹ thuật kiểm thử hộp đen để thiết kế ít nhất 50 bộ test case:



# **TÀI LIỆU THAM KHẢO**

[1] Lectures on Software Testing from Prof. Nguyễn Văn Hiệp

# **PHỤ LỤC**

# **PHÂN CÔNG CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **STT** | **Nội dung** | **Phụ trách** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |