

**TRƯỜNG ĐẠI HỌC MỞ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\_\_\_\_\_\_\_\_\_\_\_ 🙞🕮🙜 \_\_\_\_\_\_\_\_\_\_\_**

A blue book with a ring around it

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

**NHÓM 18**

**KIỂM THỬ PHẦN MỀM**

**KHOA ĐÀO TẠO ĐẶC BIỆT**

Hồ Chí Minh -\_\_/\_\_/\_\_

**TRƯỜNG ĐẠI HỌC MỞ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\_\_\_\_\_\_\_\_\_\_\_ 🙞🕮🙜 \_\_\_\_\_\_\_\_\_\_\_**

A blue book with a ring around it

Description automatically generated

***ĐỀ TÀI***

**KIỂM THỬ PHẦN MỀM**

**THI TRẮC NGHIỆM TIN HỌC**

| Giáo viên hướng dẫn | | : | *Nguyễn Trung Hậu* |
| --- | --- | --- | --- |
| Nhóm số  Thành viên | | :  : | *18*  *Nguyễn Phú Quý*  *Trần Quốc Tuấn*  *Đinh Chi Kiên* |
| Mã lớp | | : | *DH2101C* |
|  |  | | |

# LỜI CẢM ƠN

*Đầu tiên chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến Khoa Đào Tạo Đặc Biệt đã hướng dẫn giảng dạy, tạo điều kiện cho chúng em rèn luyện, học tập và phát triển bản thân trong suốt quá trình học tập tại trường.*

*Chúng em cũng xin gửi lời cảm ơn chân thành đến với thầy đã tận tâm giảng dạy chúng em trong suốt môn học Kiểm Thử Phần Mềm và đã dành thời gian sửa chữa và góp ý để chúng em hoàn thiện bài báo cáo môn học Kiểm Thử Phần Mềm.*

*Nhờ có nền tảng kiến thức vững vàng và những kinh nghiệm cộng với những lời góp ý chúng em đã có thể hoàn thành bài báo cáo môn học Kiểm Thử Phần Mềm một cách tốt nhất. Tuy nhiên, do việc thiếu kinh nghiệm thực tế về thực hiện kiểm thử phần mềm nên sẽ có những thiếu sót hay điểm sai trong bài báo cáo, chúng em mong thầy cô góp ý để chúng em có thể hoàn thiện bài báo cáo một cách tốt hơn!*

*Cuối lời, em kính chúc quý thầy luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa em xin chân thành cảm ơn.*

**Nhóm thực hiện**

# MỤC LỤC

[NHẬN XÉT CỦA GVHD 2](#_1fob9te)

[LỜI CẢM ƠN 3](#_3znysh7)

[MỤC LỤC](#_2et92p0)

[NỘI DUNG](#_4d34og8)

[CHƯƠNG 1: ĐẶC TẢ, MÔ HÌNH HOÁ YÊU CẦU VÀ GIỚI THIỆU CÁC CHỨC NĂNG CỦA ỨNG DỤNG](#_2s8eyo1)

[1.1. XÁC ĐỊNH YÊU CẦU](#_17dp8vu)

[1.1.1 Yêu cầu chức năng](#_3rdcrjn)

[1.1.2 Yêu cầu phi chức năng](#_26in1rg)

[1.2. Đặc tả yêu cầu](#_4i7ojhp)

[1.2.2.1 Phía người dùng](#_2xcytpi)

[1.2.2.1.1 Use case “Quên mật khẩu”](#_1ci93xb)

[1.2.2.2 Phía quản lý](#_23ckvvd)

[1.2.2.2.1 Use case “Quản lý”](#_ihv636)

[1.2.2.2.2 Use case “Báo cáo”](#_32hioqz)

[1.2.2.2.3 Use case “Trợ giúp”](#_1hmsyys)

[1.3. CƠ SỞ LÝ THUYẾT](#_1v1yuxt)

[1.3.1 Kiểm thử đơn vị](#_4f1mdlm)

[1.3.1.1 Khái niệm](#_2u6wntf)

[1.3.1.2 Lý do thực hiện kiểm thử đơn vị](#_19c6y18)

[1.3.1.3 Thực hiện kiểm thử đơn vị](#_3tbugp1)

[1.3.2 Kiểm thử tích hợp](#_28h4qwu)

[1.3.2.1 Khái niệm](#_nmf14n)

[1.3.2.2 Lý do thực hiện kiểm thử tích hợp](#_37m2jsg)

[1.3.2.3 Thực hiện kiểm thử tích hợp](#_1mrcu09)

[1.3.3 Kiểm thử hộp trắng](#_46r0co2)

[1.3.3.1 Khái niệm](#_2lwamvv)

[1.3.3.2 Thực hiện kiểm thử hộp trắng](#_111kx3o)

[1.3.4 Kiểm thử hộp đen](#_3l18frh)

[1.3.4.1 Khái niệm](#_206ipza)

[1.3.4.2 Thực hiện kiểm thử hộp đen](#_4k668n3)

[1.3.5 Kiểm thử system test](#_2dlolyb)

[1.3.5.1 Khái niệm](#_sqyw64)

[1.3.5.2 Thực hiện kiểm thử system test](#_3cqmetx)

[CHƯƠNG 2: KẾ HOẠCH KIỂM THỬ](#_1rvwp1q)

[2.1 Giới thiệu](#_43ky6rz)

[2.1.1 Mục tiêu](#_2iq8gzs)

[2.1.2 Mô tả dự án](#_xvir7l)

[2.1.3 Đối tượng khách hàng](#_3hv69ve)

[2.2 Chiến lược kiểm thử](#_1x0gk37)

[2.2.1 Mục tiêu kiểm thử4](#_4h042r0)

[2.2.2 Giả định kiểm thử](#_2w5ecyt)

[2.2.3 Quy tắc kiểm thử](#_1baon6m)

[2.2.4 Dữ liệu](#_3vac5uf)

[2.2.5 Phạm vi và mức độ kiểm tra](#_2afmg28)

[2.2.5.1 Exploratory Test](#_pkwqa1)

[2.2.5.2 Functional Test](#_39kk8xu)

[2.3 Chiến lược thực hiện](#_48pi1tg)

[2.3.1 Tiêu chí bắt đầu và kết thúc](#_2nusc19)

[2.3.2 Xác thực và quản lý lỗi](#_1302m92)

[2.3.3 Theo dõi lỗi và báo cáo](#_3mzq4wv)

[2.4 Quản lý kiểm thử](#_haapch)

[2.4.1 Công cụ quản lý kiểm thử](#_319y80a)

[2.4.2 Thiết kế kiểm thử](#_1gf8i83)

[2.4.3 Thiết kế thực hiện kiểm thử](#_40ew0vw)

[2.5 Thành viên và lịch trình](#_2fk6b3p)

[2.5.1 Thành viên](#_upglbi)

[2.5.2 Lịch trình](#_3ep43zb)

[CHƯƠNG 3: KỸ THUẬT KIỂM THỬ ÁP DỤNG](#_1tuee74)

3.1 Kiểm thử đơn vị

3.1.1 Test case 1

3.1.2 Test case 2

3.1.3 Test case 3

3.2 Kiểm thử cơ sỡ dữ liệu

3.2.1 Test case 4

3.2.2 Test case 5

3.2.3 Test case 6

3.3 Kiểm thử giao diện

3.3.1 Test case 7

3.3.2 Test case 8

3.4 Kiểm thử tiện dụng

3.4.1 Test case 9

[CHƯƠNG 4 – KẾT LUẬN](#_415t9al)

[4.1 Tự nhận xét và đánh giá kết quả làm được](#_2gb3jie)

[4.2 Hạn chế](#_vgdtq7)

[4.3 Hướng phát triển](#_3fg1ce0)

[TỔNG QUAN VỀ TEST CASE](#_1ulbmlt)

[BUG REPORTS](#_4ekz59m)

# NỘI DUNG

## CHƯƠNG 1: ĐẶC TẢ, MÔ HÌNH HOÁ YÊU CẦU VÀ GIỚI THIỆU CÁC CHỨC NĂNG CỦA ỨNG DỤNG

### 1.1. XÁC ĐỊNH YÊU CẦU

#### 1.1.1 Yêu cầu chức năng

Xây dựng một app (ứng dụng) cho phép:

- Đối với người dùng:

+ Người dùng phải bắt buộc tạo tài khoản để có thể sử dụng dịch vụ của app

+ Xem trang chủ màn hình chính: gồm có các thông tin về tài khoản của người dùng, điểm.

+ Xem các chuyên mục trong ứng dụng

+ Trắc Nghiệm: Gồm phần dẫn (câu dẫn) và các phương án trả lời cho sẵn. Người được hỏi tìm ra câu trả lời đúng trong nhiều phương án trả lời có sẵn.

+ Đăng nhập/Đăng xuất/Quên mật khẩu.

+ Chỉnh sửa thông tin người dùng.

- Đối với người quản lý:

* Quản lý nội dung của câu hỏi
* Quản lý nội dung của môn thi
* Quản lý thông tin của sinh viên

#### 1.1.2 Yêu cầu phi chức năng

- Giao diện trực quan, dễ dùng.

- Bảo mật tốt.

- Tốc độ xử lý nhanh chóng dưới 2s tải screen.

### 1.2 Đặc tả yêu cầu

##### 1.2.2.1 Phía người dùng

###### 1.2.2.1.1 Use case “Quên mật khẩu”

| **Function (Chức năng)** | Quên mật khẩu |
| --- | --- |
| **Description (Mô tả)** | ● Khi người dùng quên mật khẩu, chọn chức năng này để lấy lại mật khẩu |
| **Actor(s) (Đối tượng)** | Người dùng |
| **Inputs (Đầu vào)** | ● Người dùng |
| **Source** | ● SQL |
| **Outputs (Đầu ra/ kết quả)** | “Bạn vui lòng thông báo với giám thị coi thi để cấp lại mật khẩu” |
| **Destination** | Chương trình chính (hàm Main). |
| **Pre-condition (Tiền điều kiện)** | Người dùng quên mật khẩu và chọn chức năng quên mật khẩu |
| **Post-condition (Hậu điều kiện)** | Người dùng quên mật khẩu và chọn chức năng quên mật khẩu |
| **Basic Flow** | Use case bắt đầu khi người dùng chọn chức năng “Quên mật khẩu” ở Login   1. Chọn chức năng Quên mật khẩu 2. Ở giao diện “Login” thông báo: “Bạn vui lòng thông báo với giám thị coi thi để cấp lại mật khẩu” |
| **Alternative Flow** | 1. Người dùng chọn Quên mật khẩu |
| **Exception Flow** | 1. Người dùng chọn quit 2. Usecase kết thúc |

##### 1.2.2.2 Phía quản lý

###### 1.2.2.2.1 Use case “Quản lý”

| **Function (Chức năng)** | Quản lý |
| --- | --- |
| **Description (Mô tả)** | Admin muốn thêm, sửa, xóa thông tin về Câu hỏi, Môn thi, Sinh viên |
| **Inputs (Đầu vào)** | Thông tin của Câu hỏi, Môn thi, Sinh viên |
| **Source** | Được lấy dữ liệu từ cơ sở dữ liệu SQL. |
| **Outputs (Đầu ra/ kết quả)** | Thông tin có thể được thêm, sửa, xóa |
| **Destination** | Chương trình quản trị (hàm Admin). |
| **Requirements (Yêu cầu)** | Người dùng yêu cầu truy cập với tư cách Admin. |
| **Pre-condition** | Không có. |
| **Post-condition** | Không có |
| **Basic Flow** | Use case bắt đầu khi người dùng là admin chọn chức năng “Quản ký” ở trang Giảng viên.  1. Chọn Chức năng admin muốn và admin sẽ được chuyển đến giao diện của trang “Giảng viên”  2. Ở giao diện “Giảng viên”, admin sẽ được chọn biểu tượng thêm, biểu tượng xóa hoặc biểu tượng chỉnh sửa phù hợp với chức năng tương ứng  2.1 Chọn biểu tượng Thêm trong 3 Form Câu hỏi, Môn thi, Sinh viên  2.1.1 Người dùng nhập thông tin cần thêm vào trong ô TextBox ở giao diện “Giảng viên”  2.1.2 Người dùng chọn mục Lưu sau khi nhập thông tin cần thêm, hệ thống sẽ lưu thông tin vào Data  2.2 Chọn biểu tượng xóa  2.2.1 Chọn vào mục cần xóa. Thông báo xóa bộ sẽ hiện lên hỏi xem người dùng có chắc chắn muốn xóa hay không  2.2.3 Người dùng được chọn “có” hoặc “không” để xác nhận viện xóa hoặc không  2.3 Chọn biểu tượng chỉnh sửa  2.3.1 Người dùng chọn Thông tin muốn chỉnh sửa, ấn vào nút sửa rồi sửa dữ liệu muốn chỉnh sửa trong ô TextBox  2.3.2 Người dùng chọn biểu tượng đồng ý để chỉnh sửa thông tin vừa nhập  3. Hệ thống cập nhật các thay đổi của admin và hiển thị thông báo |
| **Side effects** | Không. |

###### 1.2.2.2.2 Use case “Báo cáo”

| **Function (Chức năng)** | Báo cáo kết quả |
| --- | --- |
| **Description (Mô tả)** | Admin muốn xem thông tin về Bảng điểm và Bài thi của từng Sinh viên |
| **Inputs (Đầu vào)** | Thông tin Sinh viên |
| **Source** | Được lấy dữ liệu từ cơ sở dữ liệu SQL. |
| **Outputs (Đầu ra/ kết quả)** | Thông tin chi tiết của Sinh viên làm bài thi: Mã SV, Họ Tên, Tên Môn, Mã CH, Câu trả lời, Ngày kiểm tra |
| **Destination** | Chương trình quản trị (hàm Admin). |
| **Requirements (Yêu cầu)** | Người dùng yêu cầu truy cập với tư cách Admin. |
| **Pre-condition** | Không có. |
| **Post-condition** | Không có |
| **Basic Flow** | Use case bắt đầu khi người dùng là admin chọn chức năng ở Form “Báo cáo” ở trang Giảng viên.  1. Xem Bảng Điểm: Chọn vào Bảng điểm. Hệ thống hiển thị điểm số của Sinh viên đã làm kiểm tra  2. Xem Bài Thi: Chọn vào Bài thi. Hệ thống hiển thi các kết quả mà Sinh viên đã chọn ở từng câu  3. Tìm Sinh viên: Chọn vào ô TextBox ở Mã SV. Nhập Mã SV cần tìm. Hệ thống hiển thị chi tiết thông tin sinh viên có Mã số đã nhập  4. Tìm theo Môn: Chọn vào ô lựa chọn ở Môn. Chọn môn thi cần tìm. Hệ thống hiển thi chi tiết thông tin những sinh viên đã kiểm tra môn vừa tìm |
| **Side effects** | Không. |

###### 1.2.2.2.3 Use case “Trợ giúp”

| **Function (Chức năng)** | Trợ giúp, hướng dẫn |
| --- | --- |
| **Description (Mô tả)** | Trợ giúp, hướng dẫn chi tiết |
| **Inputs (Đầu vào)** | Mục cần trợ giúp |
| **Source** | Được lấy dữ liệu từ cơ sở dữ liệu SQL. |
| **Outputs (Đầu ra/ kết quả)** | Hướng dẫn để người dùng có thể tham khảo và làm theo |
| **Destination** | Chương trình quản trị (hàm Admin). |
| **Requirements (Yêu cầu)** | Người dùng yêu cầu truy cập với tư cách Admin. |
| **Pre-condition** | Không có. |
| **Post-condition** | Không có |
| **Basic Flow** | Use case bắt đầu khi người dùng là admin chọn chức năng “Trợ giúp” ở trang Giảng Viên.  1. Chọn mục cần trợ giúp nếu có vấn đề không rõ |
| **Side effects** | Không. |

### 1.3. CƠ SỞ LÝ THUYẾT

#### 1.3.1 Kiểm thử đơn vị

##### 1.3.1.1 Khái niệm

Kiểm thử đơn vị là một loại kiểm thử mà mục tiêu của kiểm thử là kiểm tra các thành phần nhỏ nhất của chương trình có thể kiểm thử. Mục tiêu của kiểm thử đơn vị là để đảm bảo đơn vị đó hoạt động đúng chức năng được thiết kế. Kiểm thử đơn vị được thực hiện trong quá trình phát triển ứng dụng ngay sau khi đơn vị đó được hoàn thành. Kiểm thử đơn vị sẽ thực hiện kiểm thử độc lập trên đơn vị đó mà không có sự can thiệp của các đơn vị hay chức năng khác. Một đơn vị có thể một chức năng, một phương thức, một class, một object…

Trong quy trình phát triển phần mềm giai đoạn kiểm thử đơn vị là giai đoạn kiểm thử đầu tiên được thực hiện, việc kiểm thử đơn vị được thực hiện với kỹ thuật kiểm thử WhiteBox và việc kiểm thử đơn vị có thể được thực hiện bởi chính người lập trình hoặc nếu người lập trình không có thời gian thì người thực hiện kiểm thử sẽ là tester hoặc QA.

##### 1.3.1.2 Lý do thực hiện kiểm thử đơn vị

Phần lớn các kỹ sư phần mềm đều bỏ qua việc kiểm thử đơn vị để tiết kiệm thời gian phát triển ứng dụng nhưng chính vì thế sẽ dẫn tới các bug sẽ bị phát hiện trễ hơn dẫn tới chi phí sửa chữa các bug sẽ cao hơn. Kiểm thử đơn vị sớm sẽ giúp tìm kiếm các bug ở các giai đoạn đầu tiên, giúp giảm chi phí sửa chữa các bug.

1. Kiểm thử đơn vị giúp phát hiện lỗi sớm từ đó có thể dễ dàng sửa chữa với chi phí thấp.

2. Việc thực hiện kiểm thử đơn vị có thể giúp các kỹ sư phần mềm hiểu rõ hơn về đoạn code từ đó có thể trợ giúp việc sửa hay phát triển ứng dụng sau này.

3. Kiểm thử đơn vị có thể được dùng như tài liệu dự án

4. Kiểm thử đơn vị hỗ trợ việc re-use lại code, khi mang qua dự án mới chỉ cần các sự thay đổi nhỏ có thể áp dụng kiểm thử đơn vị vào dự án mới.

##### 1.3.1.3 Thực hiện kiểm thử đơn vị

- Kiểm thử đơn vị có hai loại:

+ Manual (thủ công).

**+ Automated (Tự động).**

- Kiểm thử đơn vị thường là được thực hiện một cách tự động nhưng vẫn có thể được thực hiện một cách thủ công. Thực hiện kiểm thử tự động sẽ giúp tiết kiệm nhân lực và thời gian hơn so với kiểm thử thủ công, khi thực hiện kiểm thử thủ công phải thực hiện việc kiểm thử dựa trên các tài liệu thiết kế.

- Kỹ thuật kiểm thử đơn vị:

- Các kỹ thuật bao gồm kiểm thử các đoạn mã code được tổng hợp ở dưới đây:

+ **Statement Coverage (Bao phủ dòng lệnh)**

+ Decision Coverage (Bao phủ quyết định)

+ Branch Coverage (Bao phủ các nhánh)

+ Condition Coverage (Bao phủ điều kiện)

+ Finite State Machine Coverage (Bao phủ trạng thái hữu hạn)

#### 1.3.2 Kiểm thử tích hợp

##### 1.3.2.1 Khái niệm

Kiểm thử tích hợp (Integration testing) là một giai đoạn trong việc kiểm thử phần mềm. Mỗi đơn vị của phần mềm được tích hợp lại với nhau và được kiểm thử coi là 1 đơn vị kiểm thử khi thực hiện.

Kiểm thử tích hợp xảy ra sau quá trình kiểm thử đơn vị đã hoàn thành. Kiểm thử tích hợp đảm bảo các đơn vị, các module ứng dụng tích hợp với nhau như các tài liệu thiết kế đề ra.

##### 1.3.2.2 Lý do thực hiện kiểm thử tích hợp

- Mặc dù mỗi module đều được kiểm thử đơn vị (Unit test) nhưng các lỗi vẫn còn tồn tại với các nguyên nhân sau:

+ Một đơn vị, một module được thiết kế bởi một lập trình viên với logic và kinh nghiệm sẽ khác với đơn vị hoặc module được kế bởi một lập trình viên khác với logic và kinh nghiệm khác, điều này dẫn tới các vấn đề khi tích hợp các đơn vị, module với nhau.

+ Tại thời điểm tích hợp, các module có thể bị thay đổi do change request đến từ khách hàng, việc kiểm thử này sẽ chưa được kiểm thử trong giai đoạn kiểm thử đơn vị.

+ Giao diện và module có thể chưa hoàn toàn tương thích, cơ sở dữ liệu có thể chưa tương thích với module.

+ Thiếu các xử lý lỗi.

##### 1.3.2.3 Thực hiện kiểm thử tích hợp

- Có nhiều loại hoặc cách tiếp cận khác nhau để kiểm thử tích hợp:

+ Kiểm thử tích hợp Big Bang.

+ Kiểm thử tích hợp Top-down.

+ Kiểm thử tích hợp Bottom-Up.

**+ Kiểm thử tích hợp gia tăng.**

+ Kiểm thử tích hợp Sandwich.

- Các bước thực hiện kiểm thử tích hợp.

1. Chọn mô-đun hoặc thành phần sẽ được kiểm tra.

2. Kiểm thử đơn vị.

3. Thiết kế các kịch bản thử nghiệm, trường hợp, và Script (Test Scenarios, Cases, and Scripts ).

4. Thực hiện kiểm tra theo test case đã viết.

5. Theo dõi & tái kiểm tra các lỗi ở trên.

6. Lặp lại các bước trên cho đến khi hệ thống hoàn chỉnh được kiểm tra đầy đủ.

#### 1.3.3 Kiểm thử hộp trắng

##### 1.3.3.1 Khái niệm

Kiểm thử hộp trắng là phương pháp kiểm thử phần mềm trong đó tester biết cấu trúc, nội dung, thiết kế của phần mềm. Người kiểm thử chọn đầu vào để thực hiện các bước thông qua mã và cho đầu ra đúng với yêu cầu. Kiến thức lập trình và kiến thức để thực hiện kiểm thử hộp trắng là rất quan trọng, cần thiết.

Đặc trưng:

- Kiểm thử hộp trắng là phương pháp kiểm thử phần mềm nhằm kiểm tra thuật toán, cấu trúc bên trong của sản phẩm. Đảm bảo các câu lệnh và điều kiện được thực hiện đúng.

- Kiểm thử hộp trắng được dùng khi trong kiểm thử đơn vị (unit testing) để đảm bảo code hoạt động đúng như yêu cầu, trước khi tích hợp với code đã được kiểm thử trước đó.

- Kiểm thử hộp trắng có thể được thực hiện từ khá sớm. Có thể thực hiện trong giai đoạn lập trình, phát triển để kiểm thử chức năng của các chức năng. Hoặc trong giai đoạn kiểm thử để đảm bảo các chức năng vẫn hoạt động đúng.

+ Người thực hiện kiểm thử hộp trắng là tester, dev (phải có kiến thức, kỹ năng về lập trình)

+ Khi kiểm thử hộp trắng thì người kiểm thử phải có kiến thức, kỹ năng về lập trình.

+ Kiểm thử sẽ tốn công sức, thời gian khi thành phần phần mềm (Hàm, module chức năng) lớn và phức tạp.

- Kiểm thử hộp trắng có 2 loại:

+ Kiểm thử luồng điều khiển để đảm bảo mọi đường thực thi của chương trình phần mềm cần được thực hiện đúng với thiết kế phần mềm.

Kiểm thử luồng dữ liệu để chắc chắn các biến được gán giá trị đúng, tức là ta phải xác định được đường đi của một biến từ lúc nó được khai báo tới lúc nó được sử dụng, đảm bảo nó vẫn đúng giá trị dựa trên thiết kế.

##### 1.3.3.2 Thực hiện kiểm thử hộp trắng

- Khi tiến hành làm kiểm thử bằng phương pháp kiểm thử hộp trắng thì ta cần chuẩn bị một bộ testcase cho chương trình đó. Tuy nhiên, để biết được bộ testcase của chúng ta đã đầy đủ cho tất cả các trường hợp hay chưa thì ta sẽ áp dụng các kiến thức, kỹ thuật của coverage tesing để đo đạc, tính toán các kết quả của chương trình khi thực hiện bộ kiểm thử.

- Coverage testing có thể hiểu là tỉ lệ (tính theo %) test case đã được thực hiện trên tổng số test case cần thiết cho ứng dụng. Nếu tỉ lệ này càng cao thì chương trình càng được kiểm tra kỹ càng. Mặc dù việc đảm bảo cho chương trình của chúng ta có test coverage là 100% trong một số trường hợp là bất khả thi, tuy nhiên ta vẫn cố gắng để đạt được kết quả gần với con số đó nhất.

- Có nhiều cấp độ phủ khác nhau của kiểm thử, nhưng có các loại chính sau:

+ Bao phủ chức năng (Functio coverage): Có bao nhiêu function được dịnh nghĩa trong chương trình đã được thực thi.

+ Bao phủ câu lệnh (Statement coverage): Có bao nhiêu câu lệnh trong chương trình đã được thực thi.

+ Bao phủ rẻ nhánh (Branches coverage): Có bao nhiêu nhánh trong cấu trúc điều khiển (các cấu trúc nhánh như if, switch,…) trong chương trình đã được thực thị.

+ Bao phủ điều kiện (Condition coverage): Có bao nhiêu câu điều kiện true-false đã được thực thi cả false và true.

+ Bao phủ dòng (Line coverage): Có bao nhiêu dòng code trong chương trình đã được thực thi.

+ Bao phủ điều kiện phức (Multiple condition coverage): Có bao nhiêu tổ hợp điều kiện true-false đã được thực thi.

#### 1.3.4 Kiểm thử hộp đen

##### 1.3.4.1 Khái niệm

- Kiểm thử hộp đen: là một phương pháp kiểm thử phần mềm được thực hiện mà không biết được cấu tạo bên trong của phần mềm, là cách mà các tester kiểm tra xem hệ thống như một chiếc hộp đen, không có cách nào nhìn thấy bên trong của cái hộp.

- Phương pháp này được đặt tên như vậy bởi vì các chương trình phần mềm, trong con mắt của các tester, giống như một hộp đen; bên trong mà người ta không thể nhìn thấy. Phương pháp này cố gắng tìm ra các lỗi trong các loại sau:

+ Chức năng không chính xác hoặc thiếu.

+ Lỗi giao diện.

+ Lỗi trong cấu trúc dữ liệu hoặc truy cập cơ sở dữ liệu bên ngoài.

+ Hành vi hoặc hiệu suất lỗi.

+ Khởi tạo và chấm dứt các lỗi.

- Ưu điểm của kiểm thử hộp đen chính là các tester không cần có nền tảng công nghệ, không cần biết ngôn ngữ lập trình đều có thể thực hiện test được. Người kiểm thử khi vận dụng Black box testing sẽ là một phương diện độc lập, có cái nhìn khách quan về sản phẩm.

- Phương pháp kiểm thử hộp đen sẽ cố gắng tìm ra lỗi ở các vấn đề sau: Chức năng không chính xác hoặc thiếu, lỗi giao diện, lỗi trong cấu trúc dữ liệu hoặc truy cập cơ sở bên ngoài, lỗi về hiệu suất…

##### 1.3.4.2 Thực hiện kiểm thử hộp đen

- Quy trình kiểm thử hộp đen có thể áp dụng theo 4 bước: Lập kế hoạch kiểm thử, thiết kế Test Case, thực hiện test và báo cáo kiểm thử:

1. Lập kế hoạch test. Tester tiến hành phân tích yêu cầu và lập tài liệu tổng quan về việc kiểm thử dự án bao gồm những thông tin sau:

+ Phạm vi kiểm thử, hướng tiếp cận, quy trình kiểm thử, tài nguyên và nhân lực test.

+ Các chức năng/module cần được kiểm tra; các công cụ và môi trường kiểm thử cần có.

+ Ai test chức năng nào? - Khi nào bắt đầu thực hiện viết và hoàn thành test case? - Khi nào bắt đầu thực hiện và hoàn thành test?

2. Thiết kế Test case. Sau khi có được Test Plan, Tester bắt đầu xây dựng bộ Test Case dựa trên yêu cầu của phần mềm. Test Case cần mô tả được chi tiết dữ liệu đầu vào, hành động, kết quả mong đợi để xác định một chức năng của ứng dụng phần mềm có hoạt động đúng hay không. Template của Test Case có nhiều trường hợp nhưng bắt buộc phải có 5 mục chính: ID, mục đích kiểm thử, các bước thực hiện, kết quả mong đợi & kết quả thực tế.

3. Thực hiện kiểm thử. Khi developers đã code và đưa sản phẩm lên môi trường kiểm thử, tester sẽ thực thi dựa trên Test Case đã viết. Trong quá trình test, nếu phát hiện ra bug (lỗi) thì tester sẽ log (viết) lên các tool quản lý lỗi. Bug của lập trình viên nào sẽ giao lại cho người đấy xử lý. Khi nào developers fix bug xong, tester sẽ nhận lại và tiến hành kiểm thử.

4. Báo cáo kiểm thử: Ở giai đoạn này, tester chuẩn bị báo cáo kết thúc kiểm thử, tổng hợp lại các chỉ số trong quá trình test. Cả team phát triển sẽ ngồi họp để đánh giá toàn bộ các tiêu chí xác định có thể kết thúc quy trình kiểm thử hay chưa. Những tiêu chí này khác nhau tùy theo từng dự án, thông thường bao gồm:

+ Số lượng test case tối đa được thực thi Passed.

+ Tỷ lệ lỗi giảm xuống dưới mức nhất định.

+ Deadline được chốt từ giai đoạn làm kế hoạch kiểm thử.

#### 1.3.5 Kiểm thử system test

##### 1.3.5.1 Khái niệm

System test nghĩa là test toàn bộ hệ thống. Tất cả các module/components được tích hợp theo thứ tự để xác minh rằng hệ thống làm việc đúng hay không.

##### 1.3.5.2 Thực hiện kiểm thử system test

System Test là một phần cơ bản của kiểm thử phần mềm, và test plan phải luôn chứa không gian cụ thể cho loại test này.

Để test toàn bộ hệ thống, requirement và expectation phải rõ ràng và tester cũng cần hiểu cách sử dụng thời gian thực của ứng dụng.

Ngoài ra, hầu hết các tool bên thứ ba, phiên bản OS, kiến trúc của OS có thể ảnh hưởng đến chức năng của hệ thống, hiệu năng, bảo mật, khả năng phục hồi hoặc cài đặt.

Do đó, trong khi test hệ thống, một bức tranh rõ ràng về cách sử dụng ứng dụng như thế nào, và các loại vấn đề có thể phải đối mặt trong thực tế là hữu ích. Hơn nữa, tài liệu requirement cũng quan trọng như cách hiểu ứng dụng.

Clear và update tài liệu requirement có thể giúp tester tránh được một số hiểu lầm, giả định và các câu hỏi.

## CHƯƠNG 2: KẾ HOẠCH KIỂM THỬ

### 2.1 Giới thiệu

#### 2.1.1 Mục tiêu

Kế hoạch kiểm thử mô tả phương pháp thử nghiệm và framework tổng thể quá trình thử nghiệm của PHẦN MỀM THI TRẮC NGHIỆM

Nội dung kế hoạch:

* Chiến lược kiểm tra: các quy tắc kiểm tra sẽ sử dụng, bao gồm các điều khoản nhất định của dự án (ví dụ: Câu hỏi, Kiểm tra, Làm bài); mô tả quy trình để thiết lập kiểm thử hợp lệ (ví dụ: tiêu chí input/output, các trường hợp thử nghiệm, các nhiệm vụ cụ thể để thực hiện, lập kế hoạch, chiến lược dữ liệu).
* Chiến lược thực hiện: mô tả cách kiểm thử sẽ được thực hiện và xử lý để xác định và báo cáo lỗi, cũng như để khắc phục và thực hiện sửa lỗi.
* Quản lý kiểm tra: quy trình xử lý kiểm thử và tất cả các sự kiện xảy ra trong quá trình thực hiện .

#### 2.1.2 Mô tả dự án

* PHẦN MỀM THI TRẮC NGHIỆM là một ứng dụng hoặc chương trình máy tính được thiết kế để tự động hóa quá trình thi trắc nghiệm. Thông qua phần mềm này, người dùng có thể tạo ra các bài kiểm tra trắc nghiệm, cung cấp câu hỏi và đáp án, sau đó cho phép người tham gia thi trả lời các câu hỏi này trên máy tính hoặc thiết bị di động. Phần mềm thường có tính năng tự động chấm điểm và tạo ra báo cáo kết quả sau khi thi xong. Điều này giúp giáo viên, nhà quản lý đào tạo hoặc tổ chức kiểm tra có thể quản lý và đánh giá kiến thức của người tham gia một cách hiệu quả và tiện lợi.

#### 2.1.3 Đối tượng khách hàng

* Đối tượng hướng đến là Giáo viên và Học sinh

### 2.2 Chiến lược kiểm thử

#### 2.2.1 Mục tiêu kiểm thử

Mục tiêu của thử nghiệm là để xác định và kiểm tra các chức năng của PHẦN MỀM THI TRẮC NGHIỆM hoạt động đúng với yêu cầu đặt ra.

Quá trình kiểm tra sẽ thực thi và xác minh các tập lệnh kiểm tra, xác định, sửa chữa và kiểm tra lại tất cả các lỗi theo input và output, sắp xếp theo mức độ nghiêm trọng và tính cần thiết.

Kết quả cuối cùng của quá trình kiểm thử:

* PHẦN MỀM THI TRẮC NGHIỆM
* Tập hợp các test scripts cũng như test cases trong quá trình kiểm thử để tiến hành xem xét và tái sử dụng.

#### 2.2.2 Giả định kiểm thử

Key Assumptions:

* Yêu cầu về dữ liệu và khả năng sử dụng có sẵn trước khi bắt đầu thực hiện kiểm thử chức năng.

Kiểm thử chung:

* Exploratory Testing sẽ được thực hiện sau khi build đã sẵn sàng.
* Thử nghiệm hiệu suất không được xem xét.
* Tất cả các lỗi sẽ đi kèm với một bản chụp định dạng JPEG
* Tất cả các đầu vào cần thiết được yêu cầu trong quá trình thiết kế và thực hiện kiểm thử sẽ được cung cấp.
* Các hoạt động thiết kế test case sẽ được thực hiện bởi các thành viên.

#### 2.2.3 Quy tắc kiểm thử

* Thử nghiệm sẽ tập trung vào việc đáp ứng các mục tiêu đặt ra, hiệu quả sử dụng và trải nghiệm người dùng của ứng dụng.
* Sẽ có các quy trình chung, nhất quán cho tất cả các thành viên hỗ trợ hoạt động kiểm thử.
* Các quy trình kiểm thử sẽ được xác định rõ ràng nhưng vẫn linh hoạt với khả năng thay đổi khi cần thiết.
* Các hoạt động kiểm thử sẽ được xây dựng dựa trên các giai đoạn trước để tránh dư thừa hoặc trùng lặp.
* Kiểm tra sẽ là một hoạt động có thể lặp lại, định lượng và đo lường được.
* Thử nghiệm sẽ được chia thành các giai đoạn riêng biệt, mỗi giai đoạn có mục tiêu và mục tiêu được xác định rõ ràng.
* Sẽ có tiêu chí đầu vào và đầu ra..

#### 2.2.4 Phạm vi và mức độ kiểm tra

##### 2.2.4.1 Exploratory Test

Mục đích: đảm bảo các lỗi nghiêm trọng được loại bỏ trước khi các kiểm thử tiếp theo có thể bắt đầu.

Phạm vi: application navigation, admin modules.

Tester: tất cả thành viên.

Phương pháp: Exploratory testing được thực hiện trong ứng dụng mà không có bất kỳ tập lệnh thử nghiệm nào.

Thời gian: đầu quá trình kiểm thử.

##### 2.2.4.2 Functional Test

Mục đích: kiểm tra chức năng sẽ được thực hiện để kiểm tra các chức năng của ứng dụng. Được thực hiện bằng cách cung cấp đầu vào và xác nhận đầu ra từ ứng dụng.

Phạm vi: bao gồm các chức năng chính của ứng dụng.

* Xem trang chủ màn hình chính: gồm có các thông tin về tài khoản của người dùng, điểm.
* Xem các chuyên mục trong ứng dụng.
* Trắc Nghiệm: gồm phần dẫn (câu dẫn) và các phương án trả lời cho sẵn. Người được hỏi tìm ra câu trả lời đúng nhất trong nhiều phương án trả lời có sẵn.
* Đăng nhập
* Đăng ký
* Quên mật khẩu
* Thêm câu hỏi
* Sửa câu hỏi
* Xóa câu hỏi
* Hủy câu hỏi
* Lưu câu hỏi
* Thoát
* Vào thi

Tester: tất cả thành viên.

Phương pháp: việc kiểm tra sẽ được thực hiện theo cripts tạo trước.

Thời gian: sau khi Exploratory test xong.

## 2.3 Chiến lược thực hiện

#### 2.3.1 Tiêu chí bắt đầu và kết thúc

Tiêu chí bắt đầu vào đề cập đến các điều kiện mong muốn để bắt đầu thực hiện kiểm thử.

Tiêu chí kết thúc là những điều kiện mong muốn cần phải đáp ứng để tiến hành thực hiện.

Tiêu chí bắt đầu và kết thúc là các tiêu chuẩn linh hoạt. Nếu chúng không được đáp ứng, nhóm thử nghiệm sẽ đánh giá rủi ro, xác định các hành động giảm thiểu và đưa ra giải pháp.

Tiêu chí bắt đầu để bắt đầu giai đoạn thực hiện kiểm thử, các hoạt động được liệt kê trong kế hoạch kiểm thử được hoàn thành 100%.

Tiêu chí bắt đầu để bắt đầu mỗi chu kỳ các hoạt động được liệt kê trong phần thực hiện kiểm tra của lịch trình được hoàn thành 100% ở mỗi chu kỳ.

| **Exit Criteria** | **Test Team** | **Notes** |
| --- | --- | --- |
| 100% Test case được thực thi |  |  |
| Tỷ lệ Pass 95% của Test case |  |  |
| Không có lỗi nghiêm trọng và mức độ nghiêm trọng cao |  |  |
| Tất cả các lỗi còn lại đều bị hủy bỏ hoặc  được ghi lại dưới dạng Yêu cầu thay đổi cho bản phát hành trong tương lai |  |  |
| Tất cả các kết quả mong đợi và thực tế đều được ghi lại và note với kịch bản thử nghiệm |  |  |

#### 2.3.2 Xác thực và quản lý lỗi

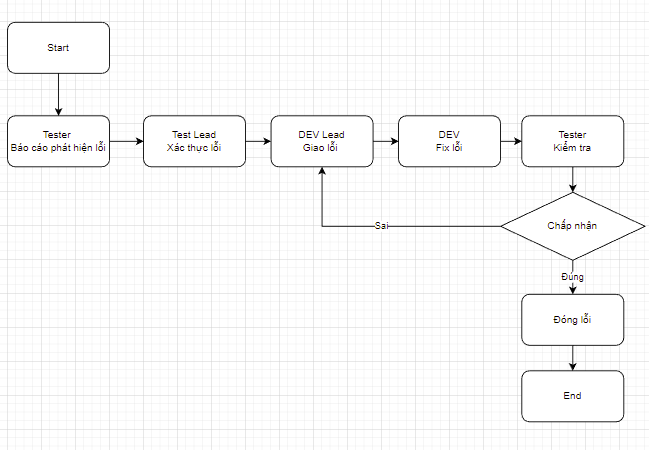
Tester sẽ thực thi tất cả các script trong mỗi chu kỳ được mô tả ở trên. Tuy nhiên, các tester cũng có thể thực hiện thử nghiệm bổ sung nếu họ xác định được một lỗ hổng có thể có trong các script. Nếu một lỗ hổng được xác định, các script sẽ được cập nhật và sau đó một lỗi được ghi lại đối với các script.

Người kiểm tra có trách nhiệm mở các lỗi, liên kết chúng với script tương ứng, chỉ định mức độ nghiêm trọng và trạng thái ban đầu, kiểm tra lại và đóng lỗi.

Mức độ nghiêm trọng của lỗi:

| Mức độ | Ảnh hưởng |
| --- | --- |
| High | Không thực hiện được các chức năng chương trình.  Lỗi có thể gây crash ứng dụng.  Không thể kết nối firebase. |
| Medium | Các chức năng có output khác với output được xác định. |
| Low | Các chức năng hoạt động không ổn định. |

#### 2.3.3 Theo dõi lỗi và báo cáo

Flowchart

*Hình 4: Flowchart*

### 2.4 Quản lý kiểm thử

#### 2.4.1 Công cụ quản lý kiểm thử

#### 2.4.2 Thiết kế kiểm thử

Tester sẽ hiểu từng yêu cầu và chuẩn bị test case tương ứng để đảm bảo tất cả các yêu cầu được bao gồm.

Mỗi test case sẽ được ánh xạ tới use case tới các yêu cầu.

Mỗi trường hợp kiểm thử sẽ được xem xét, đánh giá và các lỗi sẽ được ghi lại và chia sẻ với nhóm tester. Tester sẽ làm lại bài đánh giá khiếm khuyết và cuối cùng nhận được sự chấp thuận và phê duyệt.

Trong giai đoạn chuẩn bị, tester sẽ sử dụng nguyên mẫu, trường hợp sử dụng và chức năng

đặc điểm kỹ thuật để viết các trường hợp thử nghiệm từng bước.

#### 2.4.3 Thiết kế thực hiện kiểm thử

Sau khi tất cả các trường hợp kiểm thử được phê duyệt và môi trường kiểm thử đã sẵn sàng để kiểm thử, người kiểm thử sẽ bắt đầu Exploratory Test ứng dụng để đảm bảo ứng dụng ổn định trong thử nghiệm.

Mỗi Tester được chỉ định test case trực tiếp.

Mỗi tester thực hiện từng bước thực hiện và cập nhật trạng thái thực thi. Tester sẽ nhập trạng thái cho từng bước trực tiếp.

Nếu có bất kỳ lỗi nào, lỗi sẽ được xác định theo hướng dẫn về mức độ nghiêm với các bước để mô phỏng cùng với ảnh chụp màn hình nếu thích hợp.

Nếu có bất kỳ lỗi nào không phải là một phần của các bước nhưng có thể nằm ngoài các bước kiểm tra, những lỗi như vậy cần được ghi lại và ánh xạ nó theo cấp độ. Quá trình này được lặp lại cho đến khi tất cả các test case được thực hiện đầy đủ với trạng thái Pass/Fail.

### 2.5 Thành viên và lịch trình

#### 2.5.1 Thành viên

| **No.** | **Name** | **Role** | **Contact** |
| --- | --- | --- | --- |
| 1 | T | Tester, Dev |  |
| 2 | N | Tester |  |
| 3 | D | Tester Lead, Dev Lead, Dev |  |
| 4 | L | Tester |  |

#### 2.5.2 Lịch trình

| **Duration** | **Task** | **Actor** | **Supervisor** |
| --- | --- | --- | --- |
| 24/4-26/4/2024 | 1. Create Test Plan for project 2. Presentation of applied Testing methods in system testing (unit test, integration test, system test, acceptance test, …) | D  L | All the team members |
| 27/4 -03/5/2024 | 1. White Box Testing   - Control Flow graphing.  - Calculate the number of linearly independent paths.  - Design test case for each path.   1. Black Box Testing | D  L | All the team members |
| 4/5 - 11/5/2024 | 1. White Box Testing (cont)   - Data flow graphing.  - Test lifetime of each variable in code unit.   1. Black Box Testing (cont) | D  L | All the team members |
| 12/5 -15/5/2024 | 1. White Box Testing (cont)   Complete at least 3 unit test in source code   1. Black Box Testing (cont)   Complete at least 50 testcase | D  T  N  L | All the team members |
| 16/5 - 17/5 | 1. Automation test Appium | T  D |  |
| 18/5 - 19/5/2024 | 1. Writing bug reports for each failure case 2. Writing report | D  T  N  L | All the team members |

## CHƯƠNG 3: KỸ THUẬT KIỂM THỬ ÁP DỤNG

# 3. Thực hiện kiểm thử

## 3.1. Kiểm thử đơn vị

### 3.1.1. Test Case 1

| Test Case ID : Q\_03  Kiểm thử Tìn kiếm | |
| --- | --- |
| Mục đích kiểm tra: | Kiểm tra chức năng tìm kiếm câu hỏi tại giao diện quản lý GV |
| Chức năng kiểm tra: | Kiểm tra khả năng xử lý lỗi khi thực hiện tìm kiếm và khả năng tìm kiếm chính xác |
| Giả định: | - Khi tìm kiếm theo mã : nhập vào chữ cái: abc để kiểm tra  - Khi tìm kiếm thực hiện nhập sai mã và nhập đúng mã để kiểm tra tính đúng đắn |
| Dữ liệu dùng để kiểm tra: | - Dữ liệu đúng: Mã câu hỏi: T02, Mã môn thi: Toán  - Dữ liệu sai để kiểm tra: abc, 10000 |
| Các bước thực hiện: | 1. Click vào ô tìm kiếm câu hỏi : gõ mã câu hỏi  2. Click vào nút "Môn học" chọn theo mã môn học |
| Kết quả mong đợi: | - Nhập đúng dữ liệu cần tìm: hiển thị chính xác Hóa đơn cần tìm kiếm  - Nhập sai dữ liệu:Không hiển thị kết quả |
| Kết quả thực hiện: | Pass, Fail |
| Đề nghị, ghi chú thêm: | Cần cải thiện chức năng bắt lỗi khi nhập sai các kiểu dữ liệu khác nhau. |

### 3.1.2. Test Case 2

| Test Case ID : Q\_04  Kiểm thử Thông tin câu hỏi | |
| --- | --- |
| Mục đích kiểm tra: | Kiểm tra chức năng liên quan đến câu hỏi |
| Chức năng kiểm tra: | Kiểm tra khả năng thêm/sửa/xóa và tìm kiếm thông tin liên quan đến câu hỏi. |
| Giả định: | - Thêm câu hỏi:  · Bỏ trống tất cả các trường dữ liệu rồi thực hiện thêm  · Nhập không đầy đủ thông tin khi thêm  · Nhập đầy đủ thông tin khi thêm khách hàng.  - Sửa: chọn vào câu hỏi  · Xóa: Không chọn câu hỏi nào khi thực hiện xóa.  - Lưu : lưu và thêm thực hiện đồng thời - |
| Dữ liệu dùng để kiểm tra: | - Mã câu hỏi , mã môn học, câu hỏi , đáp án |
| Các bước thực hiện: | 1. Nhấn vào hàng hoặc cột bất kỳ của bảng câu hỏi.  2. Xóa hết tất cả thông tin được hiển thị trên textbox  3. Bấm nút "Thêm"  …. |
| Kết quả mong đợi: | - Dữ liệu thông tin của câu hỏi phải đầy đủ trước khi thực hiện Thêm/ sửa khách hàng. Thực thông tin khách câu hỏi khi nhập đúng dữ liệu và thông báo lỗi khi tìm kiếm sai thông tin |
| Kết quả thực hiện: | Pass, Fail |
| Đề nghị, ghi chú thêm: | Lỗi ràng buộc khi xóa câu hỏi |

### 3.1.3. Test Case 3

| Test Case ID : K\_02  Kiểm thử Thông tin sinh viên | |
| --- | --- |
| Mục đích kiểm tra: | Kiểm tra chức năng liên quan đến sinh viên |
| Chức năng kiểm tra: | Kiểm tra khả năng thêm/sửa/xóa và tìm kiếm thông tin liên quan đến sinh viên |
| Giả định: | - Thêm sinh viên:  · Bỏ trống tất cả các trường dữ liệu rồi thực hiện thêm  · Nhập không đầy đủ thông tin khi thêm  · Nhập đầy đủ thông tin khi thêm khách hàng.  - Sửa: chọn vào sinh viên  · Xóa: Không chọn sinh viên nào khi thực hiện xóa.  - Lưu : lưu và thêm thực hiện đồng thời |
| Dữ liệu dùng để kiểm tra: | - Mã và tên sinh viên |
| Các bước thực hiện: | 1. Nhấn vào hàng hoặc cột bất kỳ của bảng sinh viên.  2. Xóa hết tất cả thông tin được hiển thị trên textbox  3. Bấm nút "Thêm"  ….  …. |
| Kết quả mong đợi: | - Dữ liệu thông tin của sinh viên phải đầy đủ trước khi thực hiện Thêm/ sửa . Thêm đúng thông tin sinh viên khi nhập đúng dữ liệu và thông báo lỗi ràng buộc khi bị trùng |
| Kết quả thực hiện: | Pass, Fail |
| Đề nghị, ghi chú thêm: |  |

## 3.2. Kiểm thử cơ sở dữ liệu

### 3.2.1. Test Case 4

| Test Case ID : K\_02  Kiểm thử Thông tin sinh viên từ form Quản lý | |
| --- | --- |
| Mục đích kiểm tra: | Kiểm tra chức năng liên quan đến sinh viên |
| Chức năng kiểm tra: | Kiểm tra tìm kiếm sinh viên có trên database |
| Giả định: | - Nhập đúng mã số SV  - Nhập sai mã số SV |
| Dữ liệu dùng để kiểm tra: | - Mã số sinh viên |
| Các bước thực hiện: | 1. Nhấn vào quản lý  2. Nhập mã số sinh viên….  …. |
| Kết quả mong đợi: | - Dữ liệu thông tin của sinh viên phải đầy đủ |
| Kết quả thực hiện: | Pass |
| Đề nghị, ghi chú thêm: |  |

### 3.2.1. Test Case 5

| Test Case ID : Q\_01  Kiểm thử Đăng nhập | |
| --- | --- |
| Mục đích kiểm tra: | Kiểm tra chức năng đăng nhập vào hệ thống |
| Chức năng kiểm tra: | Kiểm tra dữ liệu để đăng nhập. |
| Giả định: | - Bỏ trống tên đăng nhập hoặc mật khẩu  - Nhập sai tên đăng nhập hoặc mật khẩu  - Nhập đúng tên đăng nhập và mật khẩu |
| Dữ liệu dùng để kiểm tra: | - Thông tin Username và Password : GV và SV |
| Các bước thực hiện: | 1. Điền tên đăng nhập: … 2. Điền mật khẩu: ….  3. Chọn vai trò   3. Bấm nút "Đăng nhập". |
| Kết quả mong đợi: | - Đăng nhập với nhiều tư cách  - Khi nhập sai hoặc nhập thiếu dữ liệu: Không thể đăng nhập vào hệ thống quản lý  - Khi nhâp đúng thông tin: Truy cập thành công vào hệ thống quản lý |
| Kết quả thực hiện: | Pass |
| Đề nghị, ghi chú thêm: | Cần bổ sung chức năng ghi nhớ mật khẩu |

### 3.2.2. Test Case 6

| Test Case ID : Q\_08  Kiểm thử Thi của SV | |
| --- | --- |
| Mục đích kiểm tra: | Kiểm tra thi trực tuyến |
| Chức năng kiểm tra: | Kiểm tra chức năng đặt hàng chính xác với đặc tả yêu cầu hệ thống |
| Giả định: | - Không thao tác màn hình  - Không điền câu trả lời  - Không thoát chương trình khi hết giờ |
| Dữ liệu dùng để kiểm tra: | - |
| Các bước thực hiện: | (Thực hiện trên giao diện) |
| Kết quả mong đợi: | - Tự động thoát khi hết giờ  - Không bị lỗi ràng buộc khi chưa điền câu |
| Kết quả thực hiện: | Pass |
| Đề nghị, ghi chú thêm: |  |

## 3.3. Kiểm thử giao diện

### 3.3.1. Test Case 7

| Test Case ID : Q\_02  Kiểm thử Chuyển đổi từ Form đăng nhập sang quản lý GV | |
| --- | --- |
| Mục đích kiểm tra: | Kiểm tra chuyển đổi Form đúng theo hệ thống |
| Chức năng kiểm tra: | Kiểm tra sự chuyển đổi giữa các form với nhau |
| Giả định: | - đăng nhập form đăng nhập với tư cách GV, chuyển qua form quản lý GV |
| Dữ liệu dùng để kiểm tra: | - Không cần thông tin dữ liệu |
| Các bước thực hiện: | 1. Đăng nhập với tư cách GV |
| Kết quả mong đợi: | Load ra trang giáo viên |
| Kết quả thực hiện: | Pass |
| Đề nghị, ghi chú thêm: |  |

### 3.3.2. Test Case 8

| Test Case ID : Q\_02  Kiểm thử Chuyển đổi từ Form đăng nhập sang quản lý SV | |
| --- | --- |
| Mục đích kiểm tra: | Kiểm tra chuyển đổi Form đúng theo hệ thống |
| Chức năng kiểm tra: | Kiểm tra sự chuyển đổi giữa các form với nhau |
| Giả định: | - đăng nhập form đăng nhập với tư cách SV, chuyển qua form quản lý SV |
| Dữ liệu dùng để kiểm tra: | - Không cần thông tin dữ liệu |
| Các bước thực hiện: | 1. Đăng nhập với tư cách SV |
| Kết quả mong đợi: | Load ra trang SV |
| Kết quả thực hiện: | Pass |
| Đề nghị, ghi chú thêm: |  |

## 3.4. Kiểm thử tính tiện dụng

### 3.4.1. Test Case 9

| Test Case ID : T\_01  Ẩn thông tin mật khẩu | |
| --- | --- |
| Mục đích kiểm tra: | Kiểm tra tính tiện dụng liên quan đến mật khẩu |
| Chức năng kiểm tra: | Kiểm tra khả năng ẩn khẩu khi đăng nhập |
| Giả định: | - Chưa nhập mật khẩu. Bấm nút ẩn/hiện  - Nhập mật khẩu. Sau đó bấm nút ẩn/hiện |
| Dữ liệu dùng để kiểm tra: | - Không cần thông tin dữ liệu |
| Các bước thực hiện: | 1. Nhập thông tin vào ô Mật khẩu |
| Kết quả mong đợi: | Hệ thống Ẩn theo yêu cầu của người dùng |
| Kết quả thực hiện: | Pass |
| Đề nghị, ghi chú thêm: |  |

# 4. Kết quả đạt được

## 4.1. Kết quả kiểm thử

Tổng số Test Case thực hiện: 36

Kết quả thực tế: 32 Pass

## 

# 4.2 Nhận xét, đánh giá và đề xuất

- Chương trình được xây dựng vẫn chưa đạt được mức hoàn thiện các chức năng

- Unit test còn nhiều Failed

- Test Case vẫn còn chưa bao phủ hết hệ thống chương trình

# CHƯƠNG 4 – KẾT LUẬN

### 4.1 Tự nhận xét và đánh giá kết quả làm được

Qua việc thực hiện đề tài trên, nhóm đã có những kiến thức và kỹ năng cơ bản để có thể tiến hành xây dựng và kiểm thử dự án.   
 Trong quá trình thực hiện đề tài nhóm đã đạt được những mục tiêu sau:

* Áp dụng các kiến thức đã học vào ứng dụng.
* Hiểu và áp dụng được các level của kiểm thử như:
* User acceptance test
* System test
* Unit test
* Integration test
* Các methods của kiểm thử như:
* White box test
* Black box test

### 4.2 Hạn chế

* Một số testcase chưa phù hợp với thực tế do chưa có kinh nghiệm va chạm với cái dự án thật
* Chưa kiểm thử hết tất cả các trường hợp có thể xảy ra.

### 4.3 Hướng phát triển

* Với nỗ lực của bản thân, nhóm đã cố gắng hoàn thành yêu cầu đề tài. Do thời gian và năng lực có hạn nên các testcase kiểm thử của nhóm mới chưa bảo phủ hết tới mức độ phủ cấp 4. Nhóm sẽ thiết kế ra số lượng testcase đổ có thể phủ hết các điểm quyết định và subcondition cho các hàm trong ứng dụng để thỏa mãn được tới phủ cấp 4. Nhóm cũng định hướng kiểm thử ứng dụng với Security testing và Performance testing

# BUG REPORTS

| **ID number** | **T\_04** |
| --- | --- |
| **Name** | **Test usecase Thêm Môn Thi từ Form 'Quản lý'** |
| **Summary** | **Thêm môn mới bị mã ràng buộc dữ liệu ban đầu** |

| **ID number** | **K\_02** |
| --- | --- |
| **Name** | **Kiểm thử chức năng thêm thêm xóa sinh viên** |
| **Summary** | **Xóa sinh viên bị ràng buộc dữ liệu chỉ có thể sửa sinh viên đó** |

| **ID member** | **Q\_09** |
| --- | --- |
| **Name** | **Kiểm thử chọn câu hỏi khi vào thi** |
| **Summary** | **Cick câu trước câu sau không load được dữ liệu band đầu** |