

MỤC LỤC

DANH MỤC HÌNH ẢNH.....	i
LỜI CẢM ƠN.....	1
NHẬN XÉT CỦA KHOA.....	2
CHƯƠNG 1. GIỚI THIỆU CÔNG TY THỰC TẬP	3
1. GIỚI THIỆU CÔNG TY TNHH ARI	3
2. SẢN PHẨM CỦA CÔNG TY	4
CHƯƠNG 2. NỘI DUNG THỰC TẬP	7
1. TÌM HIỂU CÔNG TY VÀ CÁC KỸ NĂNG CƠ BẢN TRONG CÔNG TY ..	7
2. NGHIÊN CỨU KỸ THUẬT	7
2.1. Các nội dung lý thuyết.....	7
2.2. Cách viết Test case	16
3. THỰC HIỆN PROJECT	17
CHƯƠNG 3. CÁC HOẠT ĐỘNG TRONG QUÁ TRÌNH THỰC TẬP.....	18
1. QUÁ TRÌNH THỰC TẬP	18
2. PHẦN THỰC HIỆN DỰ ÁN LIÊN QUAN.....	19
2.1. Nhiệm vụ chính.....	19
2.2. Thực hiện project.....	19
3. KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐÓNG GÓP	33
CHƯƠNG 4. TỔNG KẾT	34

DANH MỤC HÌNH ẢNH

<i>Hình 1.1. Logo Công ty TNHH ARI</i>	<i>3</i>
<i>Hình 1.2. Thermomix system</i>	<i>5</i>
<i>Hình 1.3. Shariot</i>	<i>5</i>
<i>Hình 1.4. Balance</i>	<i>5</i>
<i>Hình 1.5. VnEmisoft Software</i>	<i>6</i>
<i>Hình 2.1. Các loại kỹ thuật kiểm thử.....</i>	<i>10</i>
<i>Hình 2.2. Các mức độ kiểm thử.....</i>	<i>12</i>
<i>Hình 2.3. Cấu trúc file Testcase</i>	<i>16</i>
<i>Hình 3.1. Giao diện Web đăng nhập Web Member Ari.....</i>	<i>19</i>

LỜI CẢM ƠN

Em xin chân thành cảm ơn ban giám hiệu và quý thầy cô trường Đại học Khoa học đã cung cấp và bồi dưỡng cho em những kiến thức cơ bản trong suốt 4 năm ngồi trên giảng đường đại học, để từ đó em có nền tảng áp dụng vào đợt thực tập một cách hiệu quả nhất có thể.

Em xin gửi lời cảm ơn quý **Công Ty TNHH ARI Technology Joint Stock Company** đã tạo điều kiện cho em thực tập tại công ty cũng như cung cấp, bổ sung cho em những kiến thức mà em còn thiếu sót. Quý công ty đã hỗ trợ em rất nhiều trong suốt thời gian thực tập vừa qua, dạy em những kỹ năng cần thiết, truyền đạt cho em những kinh nghiệm mà các anh chị đã trải qua trong quá trình làm việc, trao đổi với khách hàng, tác phong và tinh thần trách nhiệm trước công việc được giao, những suy nghĩ tích cực. Các anh chị cũng đã tạo cho em nhiều điều kiện để ứng dụng những kiến thức đã học vào thực tế để xây dựng những dự án thực tiễn đi từ lúc ban đầu, nâng cao trình độ, cũng như tốc độ lập trình và trí thức thêm một bậc. Em xin chân thành cảm ơn.

Vì kiến thức bản thân còn hạn chế cũng như là lần đầu tiên được va chạm và tiếp xúc với thực tế, áp dụng lý thuyết vào công việc nên bản thân còn bỡ ngỡ và lúng túng, không tránh khỏi những thiếu sót, mong quý Thầy (Cô) có thể đóng góp ý kiến để em có thể hoàn thiện hơn. Em xin chân thành cảm ơn!

Hồ Văn Nhật

Huế, ngày 26 tháng 06 năm 2023

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

CHƯƠNG 1.

GIỚI THIỆU CÔNG TY THỰC TẬP

1. GIỚI THIỆU CÔNG TY TNHH ARI

Công ty Cổ Phần Công Nghệ ARI được thành lập vào năm 2018 và có trụ sở chính tại Thành phố Hồ Chí Minh, chuyên về phát triển phần mềm, chuyển đổi số và dịch vụ tư vấn IT. Công ty cung cấp một loạt các giải pháp như quản trị nguồn nhân lực doanh nghiệp (ERP), quản trị quan hệ khách hàng (CRM), quản lý chuỗi cung ứng, phân tích dữ liệu và trí tuệ nhân tạo (AI). ARI Technology có một đội ngũ chuyên gia giàu kinh nghiệm làm việc với khách hàng trong các lĩnh vực khác nhau như sản xuất, bán lẻ, chăm sóc sức khỏe và tài chính, để kể vài ví dụ.



Hình 1.1. Logo Công ty TNHH ARI

- **Tầm nhìn:** Chúng tôi là nơi những con người chuẩn mực hàng ngày nỗ lực đóng góp cho thị trường dịch vụ số những sản phẩm chuẩn mực:
 - Mục tiêu “con người chuẩn mực” luôn được đặt trên hết:
 - Song hành với mỗi chặng đường sự nghiệp (career path) là quá trình học hỏi không ngừng (training path).
 - Sự học tạo nên những con người chuẩn mực nhằm đóng góp cho thị trường dịch vụ số những sản phẩm chuẩn mực.
- **Sứ mệnh:** ARI sẽ đóng góp gì cho sự phát triển của đất nước?
 - ARI cam kết đóng góp toàn bộ tri thức, tài liệu học thuật mà trong đó có thể coi là bí mật kinh doanh cho cộng đồng, để những người ham học hỏi hàng ngày mài giũa năng lực bản thân, từ đó đóng góp cho nền công nghiệp dịch vụ số của tổ quốc Việt Nam.
 - Góp phần nhỏ bé tạo ra những sản phẩm số giúp cho những doanh nghiệp Việt Nam vươn tầm thế giới, giúp đỡ những doanh nghiệp trên thế giới tiếp cận thị trường Việt Nam một cách hiệu suất nhất.

➤ **Niềm tin:**

- Được thành lập từ năm 2018, đến nay ARI đã trở thành đối tác cung cấp giải pháp công nghệ tiên tiến cho nhiều đối tác.
- Các sản phẩm nổi bật mà chúng tôi phát triển bao gồm: Website, Mobile App, SEO, Google Ads và Facebook Advertising. Với phương châm nỗ lực hết mình để hoàn thiện sản phẩm, chúng tôi luôn cố gắng mang đến cho khách hàng những sản phẩm với trải nghiệm tốt nhất.
- Trong suốt hơn 5 năm hoạt động, ARI đã nhận được sự tin tưởng và cộng tác của nhiều khách hàng: Thermomix, Shariot, GMT Group, SoftBank, Onetech SIA... Hơn 100 khách hàng đang trải nghiệm và sử dụng sản phẩm của doanh nghiệp chúng tôi. Và chúng tôi tin rằng con số đó sẽ còn tiếp tục tăng trưởng trong những năm tiếp theo.

➤ **Giá trị cốt lõi**

- Con người chuẩn mực
- Hoàn thiện sản phẩm
- Môi trường làm việc
- Tối ưu hiệu suất
- Chia sẻ tri thức

2. SẢN PHẨM CỦA CÔNG TY

- ✓ ARI ra đời với sứ mệnh cung cấp cho thị trường các sản phẩm số chuẩn giúp cho con người và doanh nghiệp Việt Nam vươn tầm thế giới, đồng thời cũng là cầu nối để các doanh nghiệp trên thế giới tiếp cận thị trường Việt Nam một cách hiệu suất nhất.
- ✓ ARI xuất phát điểm là những người lập trình viên biết marketing kết hợp cùng với những marketer cầu toàn rất am tường công nghệ, sự phối hợp ăn ý đó sẽ luôn giúp chúng tôi có cái nhìn thông suốt từ khi thai nghén sản phẩm cho tới lúc tung ra thị trường, đồng thời cũng giúp cho khách hàng có cái nhìn bao quát hơn về con đường chuyển đổi số của doanh nghiệp.

✓ Một số sản phẩm của công ty:



Hình 1.2. Thermomix system



Hình 1.3. Shariot



Hình 1.4. Balance



Hình 1.5. VnEmisoft Software

CHƯƠNG 2.

NỘI DUNG THỰC TẬP

1. TÌM HIỂU CÔNG TY VÀ CÁC KỸ NĂNG CƠ BẢN TRONG CÔNG TY

Thời gian: 1 tuần

Nội dung: Giới thiệu về công ty, cách tổ chức của công ty

- ✓ Sinh viên được giới thiệu về công ty, ngày thành lập, niềm tin, tầm nhìn, giá trị cốt lõi, sứ mệnh, và triết lý kinh doanh.
- ✓ Cách thức tổ chức của công ty: Sinh viên được người hướng dẫn giới thiệu về quy trình làm việc, giúp sinh viên hiểu, cũng như nắm được lý thuyết để áp dụng vào dự án....
- ✓ Sinh viên được giới thiệu về cách thức làm việc trong công ty: thời gian đi làm, các quy định tuân thủ (báo công hàng ngày, tắt tiếng chuông điện thoại,...).

Kết quả:

- ✓ Hiểu được cách thức làm việc của công ty Online ARI, quy trình thành lập và phát triển.
- ✓ Biết được quy trình làm việc, cách thức cơ cấu tổ chức của công ty từ cao đến thấp
- ✓ Sinh viên nắm được việc sử dụng email (do công ty cấp), làm việc có kế hoạch, có kỷ luật, trách nhiệm hơn.

2. NGHIÊN CỨU KỸ THUẬT

2.1. Các nội dung lý thuyết

Thời gian: 1 tuần

Nội dung: Tìm hiểu về các nội dung lý thuyết sẽ được sử dụng trong quá trình làm việc.

- ✓ Trong thời gian này, anh Nguyễn Quốc đã hướng dẫn em tìm hiểu về những kiến thức quan trọng cho việc Kiểm thử phần mềm, cụ thể ở đây là kiểm thử Website.
- ✓ **Khái niệm về các kỹ thuật kiểm thử:**
 - **Kiểm thử thủ công (Manual Testing)**

Kiểm thử thủ công là kiểm thử một phần mềm một cách thủ công (không sử dụng bất kỳ công cụ tự động hoặc bất kỳ đoạn code nào). Với loại kiểm thử này, tester như người sử dụng cuối sẽ Kiểm thử phần mềm để xác định bất kỳ hành vi không mong muốn hoặc lỗi. Có rất nhiều giai đoạn để kiểm thử bằng tay: Kiểm thử đơn vị (Unit testing), Kiểm thử

tích hợp (Integration testing), Kiểm thử hệ thống (System testing) và Kiểm thử chấp nhận (User Acceptance testing).

Bất kỳ ứng dụng mới nào cũng phải được kiểm thử thủ công trước khi thực hiện kiểm thử tự động. Kiểm thử thủ công đòi hỏi nhiều nỗ lực hơn nhưng lại rất cần thiết. Kiểm thử tính khả thi, sau đó mới thực hiện kiểm thử tự động.

Tester sử dụng kế hoạch kiểm thử (test plans), trường hợp kiểm thử (test case), hoặc kịch bản kiểm thử (test scenarios) để đảm bảo tính đầy đủ của kiểm thử. Kiểm thử thủ công cũng bao gồm kiểm thử khám phá, tester kiểm thử khám phá phần mềm để tìm ra lỗi trong phần mềm đó.

Ưu điểm:

- Nhận phản hồi trực quan nhanh và chính xác.
- Ít tốn kém hơn vì không cần phải chi ngân sách cho các công cụ và quy trình tự động.
- Sự phán đoán và trực giác của con người luôn có lợi cho yếu tố thủ công.
- Trong khi kiểm thử một thay đổi nhỏ, một kiểm thử tự động sẽ yêu cầu coding làm tốn thời gian. Trong khi kiểm thử thủ công lại không tốn thời gian.

Nhược điểm:

- Phương pháp kiểm thử thủ công ít đáng tin cậy hơn vì được thực thi bởi con người. Do đó, dễ mắc sai lầm & không tìm được lỗi.
- Quá trình kiểm thử thủ công không thể được ghi lại, do đó không thể sử dụng lại.
- Một số phần nhất định khó thực hiện thủ công, có thể cần thêm thời gian.

○ **Kiểm thử tự động (Automation Testing):**

Kiểm thử tự động (Automation Testing) là khi tester viết kịch bản và sử dụng những công cụ hỗ trợ để kiểm thử phần mềm. Quá trình này là chuỗi tự động hóa quá trình kiểm thử thủ công. Kiểm thử tự động được sử dụng để chạy lại các kịch bản kiểm thử đã được kiểm thử thủ công, nhanh chóng, lặp lại nhiều lần.

Ngoài kiểm thử hồi quy, kiểm thử tự động cũng được sử dụng để Kiểm thử ứng dụng tải, performance, và stress. Nó tăng thêm độ bao phủ, cải thiện độ chính xác, tiết kiệm thời gian và chi phí so với kiểm thử thủ công

Kiểm thử tự động rất quan trọng vì những lý do sau:

- Kiểm thử thủ công tất cả các kịch bản nhiều lần sẽ gây tốn thời gian và chi phí.

- Rất khó để kiểm thử thủ công các trang web hay phần mềm đa ngôn ngữ.
- Kiểm thử tự động không cần sự can thiệp của con người. Có thể chạy kiểm thử tự động qua đêm mà không cần giám sát.
- Kiểm thử tự động làm tăng tốc độ thực hiện kiểm thử.
- Kiểm thử tự động giúp tăng phạm vi kiểm thử.
- Kiểm thử thủ công có thể trở nên nhàm chán, do đó dễ bỏ sót lỗi.

Ưu điểm:

- Hầu hết các phần của quy trình kiểm thử là tự động, bạn có thể có một quy trình nhanh chóng và hiệu quả.
- Quá trình tự động có thể được ghi lại. Điều này cho phép bạn sử dụng lại và thực hiện cùng loại hoạt động kiểm thử.
- Kiểm thử tự động được thực hiện bằng các công cụ phần mềm, do đó, có thể hoạt động liên tục mà không biết mệt mỏi.
- Có thể dễ dàng tăng năng suất, cung cấp kết quả kiểm thử nhanh & chính xác.
- Kiểm thử tự động hỗ trợ các ứng dụng khác nhau.
- Phạm vi kiểm thử có thể được tăng lên vì công cụ kiểm thử tự động hóa không bỏ qua các đơn vị kiểm thử, ngay cả đơn vị nhỏ nhất.

Nhược điểm:

- Không có yếu tố con người, thật khó để hiểu sâu hơn về các khía cạnh trực quan của giao diện người dùng như màu sắc, phông chữ, kích thước, độ tương phản hoặc kích thước nút bấm, layout...
- Các công cụ để chạy kiểm thử tự động có thể tốn kém, có thể làm tăng chi phí kiểm thử của dự án.
- Công cụ kiểm thử tự động chưa phải là bằng chứng đầy đủ. Mỗi công cụ tự động đều có những hạn chế của chúng làm giảm phạm vi tự động.
- Gỡ lỗi test scripts là một vấn đề lớn trong kiểm thử tự động. Kiểm thử bảo trì sẽ tốn kém.

✓ **Các kỹ thuật kiểm thử:**



Hình 2.1. Các loại kỹ thuật kiểm thử

○ **Phương pháp kiểm thử phần mềm White box testing:**

White box testing (Kiểm thử hộp trắng) là một kỹ thuật Kiểm thử cấu trúc bên trong của phần mềm và lấy dữ liệu kiểm thử từ logic/ mã code. Là phương pháp kiểm thử mà các chuyên gia tester tập trung vào các dữ liệu đầu vào và ra, truy cập thẳng vào bên trong source code. Các tên khác của kiểm thử hộp trắng là kiểm thử hộp mở, kiểm thử theo hướng logic hoặc kiểm thử điều khiển đường dẫn hoặc kiểm thử cấu trúc.

Ưu điểm:

- Dễ dàng tự động.
- Cung cấp các quy tắc dựa trên kỹ thuật rõ ràng cho thời điểm ngừng kiểm thử.
- Buộc các chuyên gia kiểm thử phải suy luận cẩn thận về việc test lỗi vì vậy lỗi sẽ được triệt để.

Nhược điểm:

- Khá tốn thời gian và công sức.
- Vẫn sẽ tồn tại lỗi.
- Để kiểm thử được bằng phương pháp này cần có kinh nghiệm và trình độ chuyên sâu về kiểm thử.

○ **Phương pháp kiểm thử phần mềm Black box testing:**

Black box testing (Kiểm thử hộp đen) là một phương pháp kiểm thử phần mềm Kiểm thử chức năng của ứng dụng dựa trên các đặc điểm kỹ thuật của nó. Nó còn được gọi là kiểm thử dựa trên thông số kỹ thuật.

Ưu điểm:

- Các tester khi dùng phương pháp này sẽ không cần liên quan đến code.
- Có thể tìm được nhiều bug hơn.
- Việc kiểm thử được thực hiện bởi một cách độc lập với các Dev, cho phép quan điểm khách quan và tránh sự thiên vị.

Nhược điểm:

- Chỉ có một số lượng nhỏ các đầu vào có thể được kiểm thử và nhiều đường dẫn chương trình hoặc 1 vài phần cuối sẽ không được kiểm thử.
- Các kiểm thử có thể thừa nếu nhà thiết kế/ nhà phát triển phần mềm đã chạy kiểm thử.

Vì vậy, black box testing có ưu điểm là sản phẩm phần mềm được Kiểm thử theo một quan điểm độc lập tuy vậy vẫn còn khá nhiều nhược điểm đáng lưu ý.

○ Phương pháp kiểm thử phần mềm Grey box testing:

Phương pháp Grey box testing là một trong các phương pháp kiểm thử phần mềm phổ biến nhất hiện nay. Có thể nói phương pháp Gray Box testing là phương pháp của sự kết hợp giữa White box testing và Black box testing. Kiểm thử hộp xám cho khả năng Kiểm thử cả hai mặt của một ứng dụng, lớp trình bày cũng như phần code. Nó chủ yếu là hữu ích trong kiểm thử tích hợp và kiểm thử thâm nhập. Trong kiểm thử hộp xám, cấu trúc bên trong sản phẩm chỉ được biết một phần, Tester có thể truy cập vào cấu trúc dữ liệu bên trong và thuật toán của chương trình với mục đích là để thiết kế test case, nhưng khi kiểm thử thì kiểm thử như là người dùng cuối hoặc là ở mức kiểm thử hộp đen.

Ưu điểm:

- Là sự kết hợp của kiểm thử hộp đen và hộp trắng nên sẽ tối ưu hơn.
- Kiểm thử bằng phương pháp hộp màu xám có thể thiết kế kịch bản kiểm thử phức tạp một cách thông minh hơn.

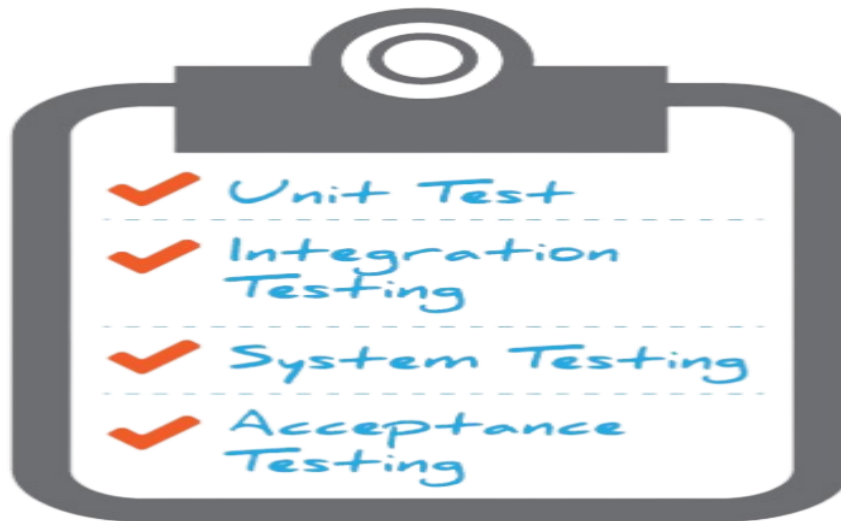
Nhược điểm:

- Rất khó để liên kết lỗi khi thực hiện kiểm thử hộp xám cho một ứng dụng có hệ thống phân tán.

Trên đây là 3 phương pháp kiểm thử phần mềm cơ bản nhất mà bất cứ một lập trình

viên nào cũng cần nắm. Việc lựa chọn phương pháp nào phụ thuộc vào khả năng cũng như dự án mà bạn đang thực hiện.

✓ **Các mức độ kiểm thử:**



Hình 2.2. Các mức độ kiểm thử

○ **Kiểm thử đơn vị (Unit Testing):**

- Unit testing, còn được gọi là Component testing, mức độ kiểm thử phần mềm này thường do Developer phụ trách, họ sẽ đi Kiểm thử các module, các hàm, các phương thức, các lớp... mà họ viết ra nhằm gia tăng sự tin cậy cho các chức năng mà mình đã viết.
- Kiểm thử đơn vị nằm trong phạm vi của kiểm thử hộp trắng (White-box testing), tức là kiểm thử code bên trong của một chức năng hoặc hệ thống để xem chức năng hoặc hệ thống đó được viết đúng chuẩn code hay chưa, đoạn code đó khi chạy hiệu năng có tốt hay không, có nhanh hay không, có tốn tài nguyên hay không...

○ **Kiểm thử tích hợp (Integration Testing):**

- Kiểm thử tích hợp là kiểm thử sự tương tác giữa các chức năng với nhau trong hệ thống và được thực hiện bởi tester. Ví dụ: sau khi đã Unit test chức năng đăng nhập và chức năng đăng ký thì ta có thể tiến hành kiểm thử tích hợp của 2 chức năng này để xem chúng có tương tác tốt với nhau hay không, sau khi đăng ký thành công thì ta có thể tiến hành đăng nhập bằng tài khoản đã đăng ký xem có thực hiện được không.

- Một ví dụ khác: Sau khi Unit test cho các chức năng con trong chức năng đăng ký môn học như đăng nhập, đăng ký tài khoản, Kiểm thử điều kiện tiên quyết, Kiểm thử học phí... thì ta Kiểm thử sự tích hợp giữa các chức năng này bằng cách tiến hành đăng ký một môn học để xem sự tương tác giữa các chức năng này có thực hiện được không, có trơn tru không, có bị mất liên kết chỗ nào không ...
 - Integration Testing có 2 cách tiếp cận phổ biến: Tích hợp từ dưới đi lên (Bottom-up Integration) và tích hợp từ trên đi xuống (Top-down Integration).
 - **Kiểm thử hệ thống (System Testing):**
 - Kiểm thử hệ thống là kiểm thử một hệ thống đã hoàn thành, đã tích hợp đầy đủ các chức năng nhằm Kiểm thử xem hệ thống phần mềm đó có đáp ứng đầy đủ các yêu cầu chức năng theo bản đặc tả yêu cầu phần mềm (SRS) hay không. Người thực hiện kiểm thử ở mức độ này thường là Tester.
 - Kiểm thử hệ thống thuộc phạm vi kiểm thử hộp đen (tức là Tester chỉ quan tâm đầu vào và kết quả mong đợi ở đầu ra mà không cần Kiểm thử code bên trong được viết như thế nào).
 - **Kiểm thử chấp nhận (Acceptance Testing):**
 - Mức độ kiểm thử phần mềm cuối cùng chính là Acceptance Test (Kiểm thử chấp nhận) – Kiểm thử xem hệ thống có đáp ứng đúng nhu cầu và mong đợi của khách hàng hay không.
 - Kiểm thử chấp nhận thường là trách nhiệm của người dùng hoặc khách hàng. Trong kiểm thử hệ thống, khách hàng sẽ Kiểm thử xem phần mềm được viết có hoạt động đúng như mong đợi của mình không, có đảm bảo tính tiện dụng, hiệu suất hoạt động có như mong đợi không, có bảo mật tốt hay không...
 - Tìm lỗi không phải là trọng tâm chính trong kiểm thử chấp nhận, vì việc tìm lỗi đã được đội Developer và Tester thực hiện trong các giai đoạn kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống rồi.
- Acceptance test gồm 2 loại kiểm thử là:
- Alpha Test, người dùng kiểm thử phần mềm ngay tại nơi phát triển phần mềm, lập trình viên sẽ ghi nhận các lỗi hoặc phản hồi, và lên kế hoạch sửa chữa.

- Beta Test, phần mềm sẽ được gửi tới cho người dùng để kiểm thử ngay trong môi trường thực, lỗi hoặc phản hồi cũng sẽ gửi ngược lại cho lập trình viên để sửa chữa. Lưu ý, không nhất thiết phải thực hiện tất cả các loại kiểm thử nêu trên. Tùy theo yêu cầu và đặc trưng của từng hệ thống, tùy vào khả năng và thời gian cho phép của dự án, khi lập kế hoạch, trưởng dự án sẽ quyết định áp dụng những loại kiểm thử nào.

✓ **Vòng đời của Bug:**

1. Tester tìm thấy bug/defect
2. Gán trạng thái cho bug: New/Mới
3. Chuyển bug sang cho Quản lý dự án để phân tích
4. Quản lý dự án quyết định xem bug có hợp lệ không
5. Nếu như lỗi không hợp lệ, trạng thái sẽ được chuyển thành "Rejected/Đã từ chối."
6. Nếu lỗi không bị rejected thì bước tiếp theo là kiểm tra xem nó có nằm trong phạm vi không. Giả sử chúng ta có một chức năng khác - chức năng email cho cùng một ứng dụng và bạn thấy có vấn đề với điều đó. Nhưng nó không nằm trong scope của lần phát hành ứng dụng lần này, trạng thái của bug đó có thể chuyển thành "Postponed/hoãn".
7. Tiếp theo, người quản lý cần xác minh xem đã có bug nào tương tự đã được tìm ra trước đó hay chưa. Nếu đã có rồi, bug này được chuyển trạng thái thành "Duplicate/trùng lặp".
8. Nếu không có vấn đề gì phát sinh trong khi dev fix bug thì bug này được chuyển sang trạng thái là "In- progress/đang tiến hành".
9. Khi code được fixed. Bug sẽ được gán trạng thái là "Fixed/đã sửa xong"
10. Tiếp theo, tester sẽ test lại phần code vừa được sửa. Nếu như các phần test cases liên quan đều passed thì bug đó được đóng lại hay được chuyển trạng thái thành "Closed". Nếu các trường hợp kiểm thử thất bại một lần nữa, lỗi được mở lại/re-opened và lại được chuyển giao sang cho dev
11. Hãy xem xét một tình huống trong lần release đầu tiên, một lỗi được tìm thấy theo thứ tự Fax đã được sửa và gán trạng thái đóng. Trong lần nâng cấp thứ hai, lỗi tương tự lại xuất hiện trở lại. Trong những trường hợp như vậy, một khiếm khuyết kín sẽ được mở lại.

✓ **Các loại lỗi thường gặp:**

- Lỗi chức năng; Lỗi giao tiếp; Lỗi thiếu lệnh; Lỗi cú pháp; Lỗi xử lý lỗi; Lỗi tính toán; Lỗi luồng điều khiển

✓ **Cách viết báo cáo 1 lỗi phần mềm:**

- 1) Bug number/ID: Số lỗi/ID giúp cho báo cáo lỗi và đề cập đến một lỗi dễ dàng hơn.
- 2) Bug Title (Tiêu đề lỗi): Tiêu đề lỗi sẽ được đọc nhiều nhất trước khi mở lỗi và đọc những phần tiếp theo.
- 3) Priority (Mức độ ưu tiên): Dựa vào mức độ nghiêm trọng của lỗi, bạn có thể đặt ra độ ưu tiên cho nó để có thể cung cấp mức độ ảnh hưởng của lỗi đến hệ thống trước xem trước hết nội dung. Tùy từng hệ thống, thường thì sẽ dựa vào chỉ số này để ưu tiên các bug nào sẽ phải được/nên được fix trước
- 4) Description (Mô tả): Mô tả lỗi giúp đội phát triển hiểu được lỗi. Nó mô tả các vấn đề đang gặp phải.
- 5) Step (Các bước để tái hiện): Một báo cáo lỗi tốt nên đề cập rõ đến các bước để tái hiện lỗi. Các bước nên bao gồm các hành động gây ra lỗi.
- 6) Environment (Môi trường test): Môi trường test và cấu hình trình duyệt là 1 yếu tố cần thiết cho một báo cáo lỗi rõ ràng.
- 7) Expected and Actual Result (Kết quả mong đợi và thực tế): Mô tả lỗi sẽ không đầy đủ nếu không có kết quả dự kiến và thực tế. Cần nêu được kết quả cụ thể sau khi tiến hành các bước kiểm thử là gì và mong đợi kết quả mới như thế nào.
- 8) Attachment (Bằng chứng: Video, Ảnh chụp màn hình lỗi): Chụp ảnh màn hình, quay video ở trường hợp thất bại với phụ đề phù hợp để làm rõ hơn lỗi đang báo cáo là gì, giúp người đọc có thể kiểm tra và làm theo dễ dàng.

✓ **Thực hiện:**

- Tham gia đầy đủ các buổi training
- Tiến hành tìm hiểu, nghiên cứu các nội dung.

✓ **Kết quả:**

- Có được những kiến thức quan trọng cho việc kiểm thử phần mềm sau này.
- Ngoài ra còn được biết thêm một số quy tắc trong việc viết Test Report sao cho đúng chuẩn, dễ đọc, dễ hiểu và một số quy trình phát triển phần mềm.

- Được người hướng dẫn kiểm tra lại các nội dung đã hướng dẫn

2.2. Cách viết Test case

Thời gian: 1 tuần

Nội dung: Người hướng dẫn đã hướng dẫn em thế nào là kịch bản kiểm thử, những thành phần có trong 1 Testcase và các lên 1 Testcase với 1 chức năng.

✓ Thế nào là Testcase

- Testcase: Test case là mô tả một dữ liệu đầu vào (input), hành động (action) hoặc một sự kiện (event) và kết quả truy vấn (expected response). Test case nhằm kiểm tra từng chức năng của ứng dụng phần mềm hoạt động đúng hay không.
- Test case là rất quan trọng trong bất kì dự án nào vì đây là bước đầu tiên trong quá trình test và nếu có gì đó sai sót ở bước này sẽ kéo theo hệ quả ở các giai đoạn tiếp theo trong vòng đời test.
- Một tester luôn phải biết dữ liệu nào cần thực hiện test, thứ tự test đây là điều kiện quyết định cho việc test
- Test case liệt kê yêu cầu của khách hàng. Là việc quan trọng để xác định những thay đổi mà khách hàng mong muốn. Khi thay đổi một số chức năng của việc test nó cũng không thay đổi chức năng của phần mềm hay ứng dụng.
- Một bản test case bắt buộc phải có từng trạng thái riêng để người trưởng nhóm biết được chức năng nào của ứng dụng đã được test và chưa được test, những chức năng nào nhiều lỗi và ngược lại.

✓ Cấu trúc

Testcase ID	Summary	Preconditions	Test Step	Data Test	Expected Result	Actual Result	Status (Passed/Failed)	Automation Test (Yes/No/Finished)	Written By	Notes
Verify validation text										
1	Verify that login without entering username	Go to: http://stag.balance.ari.com.vn/#auth/login	1. Go to: http://stag.balance.ari.com.vn/#auth/login 2. Enter password 3. Click 'Login'	username: "" password: "text_pass"	Show validation text at Username: "This is a required field!"	Show validation text at Username: "This is a required field!"	Passed	Yes	Mr. Nhật	
2	Verify that login without entering password	Go to: http://stag.balance.ari.com.vn/#auth/login	1. Go to: http://stag.balance.ari.com.vn/#auth/login 2. Enter username 3. Click 'Login'	username: "text_user@gmail.com" password: ""	Show validation text at Password: "This is a required field!"	Show validation text at Password: "This is a required field!"	Passed	Yes	Mr. Nhật	
3	Verify that login without entering both username and password	Go to: http://stag.balance.ari.com.vn/#auth/login	1. Go to: http://stag.balance.ari.com.vn/#auth/login 2. Click 'Login'	username: "" password: ""	Show validation text at Username and Password: "This is a required field!"	Show validation text at Username and Password: "This is a required field!"	Passed	Yes	Mr. Nhật	
4	Verify that login with username in the invalid email format	Go to: http://stag.balance.ari.com.vn/#auth/login	1. Go to: http://stag.balance.ari.com.vn/#auth/login 2. Enter username 3. Enter password 4. Click 'Login'	Email: -text_user	Show validation text at Username: "Please enter a valid email address!"	Show validation text at Username: "Please enter a valid email address!"	Passed	Yes	Mr. Nhật	

Hình 2.3. Cấu trúc file Testcase

- Testcase ID: Mã ID của testcase
- Summary: Mô tả chức năng cho việc kiểm tra chức năng đó

- Precondition: Tiền điều kiện thực hiện chức năng đó
- Test Step: Mô tả từng bước để thực hiện kiểm tra chức năng đó
- Data test: dữ liệu cần test.
- Expected Result: Kết quả mong đợi sau khi thực hiện kiểm tra chức năng đó.
- Actual Result: Kết quả thực tế sau khi thực hiện kiểm tra chức năng đó.
- Việc mô tả từng chức năng sẽ cho biết chức năng nào đang được kiểm tra và các bước thực hiện để kiểm tra chức năng đó, sau cùng là kết quả sau khi thực hiện chức năng đó mà mình mong đợi để đáp ứng cho chức năng đó.
- Status (Passed/Failed): Kết quả đầu ra của testcase Passed hoặc Failed.
- Automation Test (Yes/No/Finished): Kiểm thử tự động hay chưa.
- Written by: Người viết Test case
- Notes: ghi chú

✓ **Thực hiện:**

- Tham gia đầy đủ các buổi training
- Tiến hành tìm hiểu, nghiên cứu các nội dung.
- Thực hiện lên testcase cho các chức năng.
- Tiến hành test theo kịch bản testcase

✓ **Kết quả:**

- Hiểu được những khái niệm cơ bản về testcase.
- Biết được quy tắc viết testcase, kiểm tra các giá trị đầu vào của testcase
- Hiểu được luồng xử lý của chức năng thực hiện testcase

3. THỰC HIỆN PROJECT

✓ **Nội dung:**

- Sau gần 3 tuần được hướng dẫn, training và thực hành. Em đã nắm được những kiến thức cơ bản về kiểm thử phần mềm. Trong 1 tuần tiếp theo, anh Đức đã hướng dẫn em áp dụng những kiến thức đã học để thực hiện kiểm thử trên một dự án của công ty. Chi tiết sẽ được nói ở phần sau.

CHƯƠNG 3. CÁC HOẠT ĐỘNG TRONG QUÁ TRÌNH THỰC TẬP

1. QUÁ TRÌNH THỰC TẬP

Thời gian	Công việc	Người hướng dẫn	Mức độ hoàn thành	Nhận xét của người hướng dẫn
Tuần 1 (5/6 – 9/6/2023)	<ul style="list-style-type: none">- Tìm hiểu về công ty, cách tổ chức của công ty.- Làm quen với các công cụ làm việc trong công ty.- Học cách trao đổi, làm việc qua email.	Anh Nguyễn Quốc	100%	Tốt
Tuần 2 (12/6 – 16/6/2023)	<ul style="list-style-type: none">- Tìm hiểu quy trình hoạt động của onsite review.- Tìm hiểu các công cụ để sử dụng.- Tìm hiểu về các khái niệm, nhưng công cụ hỗ trợ cho việc kiểm thử phần mềm- Cách báo cáo khi gặp lỗi của phần mềm.	Anh Nguyễn Quốc	100%	Tốt
Tuần 3 (19/6 – 23/6/2023)	<ul style="list-style-type: none">- Viết Test case để kiểm thử Web Blance.- Tiến hành kiểm thử theo kịch bản đã dựng.	Anh Nguyễn Quốc	100%	Tốt
Tuần 4 (26/6 – 30/6/2022)	<ul style="list-style-type: none">- Viết automation test theo kịch bản đã dựng.	Anh Nguyễn Ngọc Đức	100%	Tốt

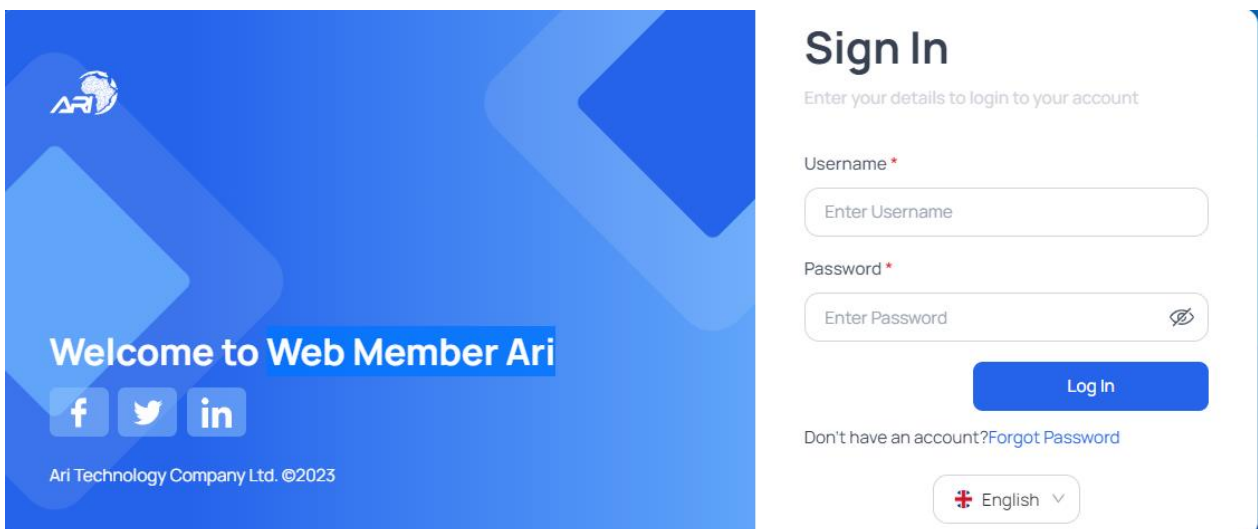
2. PHẦN THỰC HIỆN DỰ ÁN LIÊN QUAN

2.1. Nhiệm vụ chính

Được hướng dẫn	<ol style="list-style-type: none">1. Tìm hiểu về dự án và yêu cầu sản phẩm2. Tìm hiểu các kiến thức cơ bản của công việc kiểm thử phần mềm3. Luyện tập với thiết bị thực tế4. Học cách chạy các test case5. Tìm hiểu cách thiết kế test case6. Tìm hiểu cách sử dụng trang web quản lí test case7. Cách quản lý bug từ lúc phát hiện đến khi được kiểm thử và đóng quy trình.
Áp dụng vào thực tế	<ol style="list-style-type: none">1. Viết test case trong project.2. Viết automation test từ test case vừa viết trong project.

2.2. Thực hiện project

- ✓ Thực hành viết các test case với tính năng đăng nhập trên nền tảng Web Member Ari:



Hình 3.1. Giao diện Web đăng nhập Web Member Ari

- ✓ Viết test case cho tính năng đăng nhập trên nền tảng Web Member Ari:

Testcase ID	Summary	Preconditions	Test Step	Data Test	Expected Result	Actual Result	Status (Passed/Failed)	Automation Test (Yes/No/Finished)
Verify validation text:								
1	Verify that validation text of "Username" field display when logging in with "Username" field empty	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter Username 2. Click "Log In"	Password: "text_pass"	Show validation text at "Username" field: "Please enter username"		Passed	Yes
2	Verify that validation text of "Password" field display when logging in with "Password" field empty	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter Password 2. Click "Log In"	Username: "text_user@gmail.com"	Show validation text at "Password" field: "Please enter password"		Passed	Yes
3	Verify that validation text of "Username" field and "Password" field display when logging in with "Username" field and "Password" empty	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Click "Log In"		- Show validation text at "Username" field: "Please enter username" - Show validation text at "Password" field: "Please enter password"		Passed	Yes
4	Verify that validation text of "Username" field display when logging in with "Username" field invalid email format.	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter Username 2. Enter Password 3. Click "Log In"	Username: "text_user" Password: "text_pass"	Show validation text at "Username" field: "Please enter a valid email address!"		Passed	Yes
5	Verify that the authentication text of the "Username" field is displayed when logging in with a username that is less than 6 characters long	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter Username 2. Enter Password 3. Click "Log In"	Username: "text" Password: "text_pass"	Show validation text at "Username" field: "Please enter a valid email address!" "Please enter at least 6 characters!"		Passed	Yes
Verify error message								
6	Verify that error message display when logging in with "Username" field invalid and "Password" valid	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter invalid Username 2. Enter valid Password 3. Click "Log In"	Username: "text_user@gmail.com" Password: "Password1"	Show error message: "User text_user@gmail.com not found!"		Passed	Yes
7	Verify that error message display when logging in with "Username" field valid and "Password" invalid	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter valid Username 2. Enter invalid Password 3. Click "Log In"	Username: "admin@admin.com" Password: "text_pass"	Show error message: "Invalid credentials for user admin@admin.com"		Passed	Yes
8	Verify that error message display when logging in with "Username" field invalid and "Password" invalid	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter invalid Username 2. Enter invalid Password 3. Click "Log In"	Username: "text_user@gmail.com" Password: "text_pass"	Show error message: "User text_user@gmail.com not found!"		Passed	Yes
9	Verify that the error message displays when logging with a password less than 6 characters.	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter invalid Username 2. Enter invalid Password 3. Click "Log In"	Username: "text_user@gmail.com" Password: "text"	Show error message: "password must be longer than or equal to 6 characters"		Passed	Yes
Verify Sign in successfully								
10	Verify that Log In is successful when logging in with administrator account	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter valid Username 2. Enter valid Password 3. Click "Log In"	Username: "admin@admin.com" Password: "Password1"	- Navigate to Admin page - Display pop-up with successful message "Success" - Menu displays options as below: + Admin + User + Log Off + Setting		Passed	Yes
11	Verify that Log In is successful when logging in with Manager account	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter valid Username 2. Enter valid Password 3. Click "Log In"	Username: "manager@gmail.com" Password: "Nhat@01101999"	- Navigate to Admin page - Display pop-up with successful message "Success" - Menu displays options as below: + User + Log Off + Setting		Passed	Yes
12	Verify that Log In is successful when logging in with Staff account	Go to: http://duv23hk3u3v8m.cloudfront.net/vn/auth/Log_In Select language: English	1. Enter valid Username 2. Enter valid Password 3. Click "Log In"	Username: "staff@gmail.com" Password: "Nhat@01101999"	- Navigate to Admin page - Display pop-up with successful message "Success" - Menu displays options as below: + Admin + User + Log Off + Setting		Passed	Yes

Testcase ID	Summary	Preconditions	Test Step	Data Test	Expected Result	Actual Result	Status (Passed/Failed)	Automation Test (Yes/No/Finished)
Verify navigate to Forgot password page								
13	Verify that CAN navigate to the "Forgot Password" page from the link on the Log In page	Go to: http://duv23hk3u3v8m.cloudfront.net/vm/auth/Log_In Select language: English	1. Click "Forgot Password"		Navigate to Forgot Password page		Passed	Yes
14	Verify that validation text of "Email" field display when Forgot Password with "Email" field empty	Go to: http://duv23hk3u3v8m.cloudfront.net/vm/auth/Log_In Select language: English Navigate to Forgot Password page	1. Click "Send"		Show validation text at "Email" field: "Please enter email"		Passed	Yes
15	Verify that validation text of "Email" field display when Forgot Password with "Email" field invalid email format and less than 6 characters.	Go to: http://duv23hk3u3v8m.cloudfront.net/vm/auth/Log_In Select language: English Navigate to Forgot Password page	1. Enter email 2. Click "Send"	Email text	Show validation text at "Email" field: "Please enter a valid email address!" "Please enter at least 6 characters!"		Passed	Yes
16	Verify that validation text of "Email" field display when Forgot Password with "Email" field invalid email format and greater than 6	Go to: http://duv23hk3u3v8m.cloudfront.net/vm/auth/Log_In Select language: English Navigate to Forgot Password page	1. Enter email 2. Click "Send"	Email text_user	Show validation text at "Email" field: "Please enter a valid email address!"		Passed	Yes
17	Verify that the forgot password can be cancel using the "Cancel" button	Go to: http://duv23hk3u3v8m.cloudfront.net/vm/auth/Log_In Select language: English Navigate to Forgot Password page	1. Click "Cancel"		Navigate to Login page		Passed	Yes
Verify displays the password								
18	Verify that the password can be displayed in text format when clicking on the "Eye" icon.	Go to: http://duv23hk3u3v8m.cloudfront.net/vm/auth/Log_In Select language: English	1. Enter password 2. Click "Eye" icon	Username: "tex_user@gmail.com" Password: "text_password"	Passwords are displayed as characters		Passed	Yes
Verify refresh page								
19	Verify entered data not showing when Refresh button is clicked	Go to: http://duv23hk3u3v8m.cloudfront.net/vm/auth/Log_In Select language: English	1. Enter username 2. Enter password 3. Reload page with F5 key		The page is refreshed with empty fields		Passed	Yes

✓ Viết automation test cho các test case:

```
/// <reference types="cypress" />
/// <reference types="cypress-xpath" />
describe('Verify validation text', () => {
  beforeEach(()=>{
    cy.visit('http://duv23hk3u3y8m.cloudfront.net/en/auth/login')
    cy.xpath('/html/body/div[1]/div/div/div[2]/div[4]/label/div').click()
    cy.xpath('/html/body/div[2]/div/div/div[2]/div[1]/div/div/div[1]/div').click()
  })

  it('Verify that validation text of "Username" field display when logging in with "Username" field empty', () => {
    cy.get('#password').type('text_pass')
    cy.get('#idSubmit').click()
    cy.wait(2000)
    cy.get('#email_help > .ant-form-item-explain-error').should('have.text', 'Please enter username')
  })

  it('Verify that validation text of "Password" field display when logging in with "Password" field empty', () => {
    cy.get('#email').type('text_user@gmail.com')
    cy.get('#idSubmit').click()
    cy.wait(2000)
    cy.get('#password_help > .ant-form-item-explain-error').should('have.text', 'Please enter password')
  })

  it('Verify that validation text of "Username" field and "Password" field display when logging in with "Username" field and "Password" empty', () => {
    cy.get('#idSubmit').click()
    cy.wait(2000)
    cy.get('#email_help > .ant-form-item-explain-error').should('have.text', 'Please enter username')
    cy.get('#password_help > .ant-form-item-explain-error').should('have.text', 'Please enter password')
  })
})
```

```

    it('Verify that validation text of "Username" field display when logging in with
"Username" field invalid email format.', () => {
      cy.get('#email').type('text_user')
      cy.get('#password').type('text_pass')
      cy.get('#idSubmit').click()
      cy.wait(2000)
      cy.get('#email_help > .ant-form-item-explain-error').should('have.text', 'Please
enter a valid email address!')
    })

    it('Verify that the authentication text of the "Username" field is displayed when
logging in with a username that is less than 6 characters long.', () => {
      cy.get('#email').type('text')
      cy.get('#password').type('text_pass')
      cy.get('#idSubmit').click()
      cy.wait(2000)
      cy.get('#email_help > :nth-child(1)').should('have.text', 'Please enter a valid
email address!')
      cy.get('#email_help > :nth-child(2)').should('have.text', 'Please enter at least
6 characters!')
    })
  })

describe('Verify error message', () => {
  beforeEach(()=>{
    cy.visit('http://duv23hk3u3y8m.cloudfront.net/en/auth/login')
    cy.xpath('/html/body/div[1]/div/div/div[2]/div[4]/label/div').click()
    cy.xpath('/html/body/div[2]/div/div/div[2]/div[1]/div/div/div[1]/div').click()
  })

  it('Verify that error message display when logging in with "Username" field invalid
and "Password" valid', () => {
    const username = "text_user@gmail.com"
    const password = "Password1!"
    const error_message = 'User' + ' ' + username + ' not found!'
    cy.get('#email').type(username)
    cy.get('#password').type(password)
    cy.get('#idSubmit').click()
  })

```

```

    cy.wait(2000)
    cy.get('#swal2-title').should('have.text', 'Fail')
    cy.xpath('/html/body/div[3]/div/div[2]').should('have.text', error_message)
  })

  it('Verify that error message display when logging in with "Username" field invalid
and "Password" valid', () => {
    const username = "admin@admin.com"
    const password = "text_pass"
    const error_message = 'Invalid credentials for user' + ' ' + username
    cy.get('#email').type(username)
    cy.get('#password').type(password)
    cy.get('#idSubmit').click()
    cy.wait(2000)
    cy.get('#swal2-title').should('have.text', 'Fail')
    cy.xpath('/html/body/div[3]/div/div[2]').should('have.text', error_message)
  })

  it('Verify that error message display when logging in with "Username" field invalid
and "Password" invalid', () => {
    const username = "text_user@gmail.com"
    const password = "text_pass"
    const error_message = 'User' + ' ' + username + ' not found!'
    cy.get('#email').type(username)
    cy.get('#password').type(password)
    cy.get('#idSubmit').click()
    cy.wait(2000)
    cy.get('#swal2-title').should('have.text', 'Fail')
    cy.xpath('/html/body/div[3]/div/div[2]').should('have.text', error_message)
  })

  it('Verify that the error message displays when logging with a password less than 6
characters.', () => {
    const username = "text_user@gmail.com"
    const password = "text"
    const error_message = 'password must be longer than or equal to 6 characters'
    cy.get('#email').type(username)
    cy.get('#password').type(password)

```

```

    cy.get('#idSubmit').click()
    cy.wait(2000)
    cy.get('#swal2-title').should('have.text', 'Fail')
    cy.xpath('/html/body/div[3]/div/div[2]').should('have.text', error_message)
  })
})

describe('Verify Sign in successfully', () => {
  beforeEach(()=>{
    cy.visit('http://duv23hk3u3y8m.cloudfront.net/en/auth/login')
    cy.xpath('/html/body/div[1]/div/div/div[2]/div[4]/label/div').click()
    cy.xpath('/html/body/div[2]/div/div/div[2]/div[1]/div/div/div[1]/div').click()
  })

  it('Verify that Login is successful with administrator account', () => {
    const username = "admin@admin.com"
    const password = "Password1!"
    const Notification = "Success"
    cy.get('#email').type(username)
    cy.get('#password').type(password)
    cy.get('#idSubmit').click()
    cy.wait(2000)
    cy.get('#swal2-title').should('have.text', Notification)
    cy.xpath('/html/body/div[3]/div/div[2]').should('have.text', Notification)
    cy.wait(2000)
    cy.xpath('/html/body/div[1]/main/div[2]').should('have.text', 'Admin')
    cy.xpath('/html/body/div/main/div[2]/div[2]').click()
    cy.wait(2000)
    cy.xpath('/html/body/div/main/div[3]/ul/li[1]/div').should('have.text', 'User')
    cy.xpath('/html/body/div[1]/main/div[3]/ul/li[2]/div/div/div[1]').should('have.text', 'Day Off')
    cy.xpath('/html/body/div[1]/main/div[3]/ul/li[3]/div/div/div').should('have.text', 'Setting')
  })

  it('Verify that Login is successful with Manager account', () => {
    const username = "manager@gmail.com"
    const password = "Nhat@01101999"

```

```

const Notification = "Success"
cy.get('#email').type(username)
cy.get('#password').type(password)
cy.get('#idSubmit').click()
cy.wait(2000)
cy.get('#swal2-title').should('have.text', Notification)
cy.xpath('/html/body/div[3]/div/div[2]').should('have.text', Notification)
cy.wait(2000)
cy.xpath('/html/body/div[1]/main/div[2]').should('have.text', 'Admin')
cy.xpath('/html/body/div/main/div[2]/div[2]').click()
cy.wait(2000)
cy.xpath('/html/body/div/main/div[3]/ul/li[1]/div').should('have.text', 'User')
cy.xpath('/html/body/div[1]/main/div[3]/ul/li[2]/div/div/div[1]').should('have.te
xt', 'Day Off')
    cy.xpath('/html/body/div[1]/main/div[3]/ul/li[3]/div/div/div').should('have.text'
, 'Setting')
  })

it('Verify that Login is successful with Staff account', () => {
  const username = "staff@gmail.com"
  const password = "Nhat@01101999"
  const Notification = "Success"
  cy.get('#email').type(username)
  cy.get('#password').type(password)
  cy.get('#idSubmit').click()
  cy.wait(2000)
  cy.get('#swal2-title').should('have.text', Notification)
  cy.xpath('/html/body/div[3]/div/div[2]').should('have.text', Notification)
  cy.wait(2000)
  cy.xpath('/html/body/div[1]/main/div[2]').should('have.text', 'Admin')
  cy.wait(2000)
  cy.get('.hamburger').click()
  cy.wait(2000)
  cy.xpath("//span[text()='Day Off']").eq(1).should('have.text', 'Day Off')
  cy.xpath("//span[text()='Setting']").should('have.text', 'Setting')
  cy.wait(2000)
  })
})

```

```

describe('Verify navigate to Forgot password page', () => {
  beforeEach(()=>{
    cy.visit('http://duv23hk3u3y8m.cloudfront.net/en/auth/login')
    cy.xpath('/html/body/div[1]/div/div/div[2]/div[4]/label/div').click()
    cy.xpath('/html/body/div[2]/div/div/div[2]/div[1]/div/div/div[1]/div').click()
  })

  it('Verify that CAN navigate to the "Forgot Password" page from the link on the Log
In page', () => {
    cy.get('.text-blue-600').click()
    cy.wait(2000)
    cy.xpath("//h3[text()='Forgot Password']").should('have.text','Forgot Password')
  })

  it('Verify that validation text of "Email" field display when Forgot Password with
"Email" field empty', () => {
    cy.get('.text-blue-600').click()
    cy.wait(2000)
    cy.xpath("//button[text()='Send']").click()
    cy.wait(2000)
    cy.get('.ant-form-item-explain-error').should('have.text','Please enter email')
  })

  it('Verify that validation text of "Email" field display when Forgot Password with
"Email" field invalid email format and less than 6 characters.', () => {
    const email = "text"
    cy.get('.text-blue-600').click()
    cy.wait(2000)
    cy.xpath("//input[@placeholder='Enter email']").type(email)
    cy.xpath("//button[text()='Send']").click()
    cy.wait(2000)
    cy.get('#email_help > :nth-child(1)').should('have.text','Please enter a valid
email address!')
    cy.get('#email_help > :nth-child(2)').should('have.text','Please enter at least 6
characters!')
  })
})

```



```

    it('Verify that validation text of "Email" field display when Forgot Password with
"Email" field invalid email format and greater than 6 characters.', () => {
        const email = "text_user"
        cy.get('.text-blue-600').click()
        cy.wait(2000)
        cy.xpath("//input[@placeholder='Enter email']").type(email)
        cy.xpath("//button[text()='Send']").click()
        cy.wait(2000)
        cy.get('.ant-form-item-explain-error').should('have.text', 'Please enter a valid
email address!')
    })

    it('Verify that the forgot password can be cancel using the "Cancel" button', () =>
{
        const email = "text_user"
        cy.get('.text-blue-600').click()
        cy.wait(2000)
        cy.xpath("//input[@placeholder='Enter email']").type(email)
        cy.xpath("//button[text()='Cancel']").click()
        cy.xpath("//h2[@class='-intro-x font-bold text-3xl text-white mb-
3']").should('have.text', 'Welcome to Web Member Ari')
        cy.xpath('//h1').should('have.text', 'Sign In')
    })
})

describe('Verify displays the password', () => {
    beforeEach(()=>{
        cy.visit('http://duv23hk3u3y8m.cloudfront.net/en/auth/login')
        cy.xpath('/html/body/div[1]/div/div/div[2]/div[4]/label/div').click()
        cy.xpath('/html/body/div[2]/div/div/div[2]/div[1]/div/div/div[1]/div').click()
    })

    it('Verify that the password can be displayed in text format when clicking on the
"Eye" icon.', () => {
        const username = "text_user@gmail.com"
        const password = "text_password"
        cy.get('#email').type(username)
        cy.get('#password').type(password)
    })
}

```

```

    cy.wait(2000)
    cy.get('#password').invoke('attr', 'type', 'text')
  })
})

describe('Verify refresh page', () => {
  beforeEach(()=>{
    cy.visit('http://duv23hk3u3y8m.cloudfront.net/en/auth/login')
    cy.xpath('/html/body/div[1]/div/div/div[2]/div[4]/label/div').click()
    cy.xpath('/html/body/div[2]/div/div/div[2]/div[1]/div/div/div[1]/div').click()
  })
  it('Verify entered data not showing when Refresh button is clicked', () => {
    cy.get('#email').type('text_user@gmail.com')
    cy.get('#password').type('text_password')
    cy.reload()
    cy.get('#email').should("be.empty")
    cy.get('#password').should("be.empty")
  })
})

```

✓ Kết quả kiểm thử:

- ▼ Verify validation text
 - ✓ Verify that validation text of "Username" field display when logging in with "Username" field empty
 - ✓ Verify that validation text of "Password" field display when logging in with "Password" field empty
 - ✓ Verify that validation text of "Username" field and "Password" field display when logging in with "Username" field and "Password" empty
 - ✓ Verify that validation text of "Username" field display when logging in with "Username" field invalid email format.
 - ✓ Verify that the authentication text of the "Username" field is displayed when logging in with a username that is less than 6 characters long.
- ▼ Verify error message
 - ✓ Verify that error message display when logging in with "Username" field invalid and "Password" valid
 - ✓ Verify that error message display when logging in with "Username" field invalid and "Password" valid
 - ✓ Verify that error message display when logging in with "Username" field invalid and "Password" invalid
 - ✓ Verify that the error message displays when logging with a password less than 6 characters.

- ✓ Verify Sign in successfully
 - ✓ Verify that Login is successful with administrator account
 - ✓ Verify that Login is successful with Manager account
 - ✓ Verify that Login is successful with Staff account
- ✓ Verify navigate to Forgot password page
 - ✓ Verify that CAN navigate to the "Forgot Password" page from the link on the Log In page
 - ✓ Verify that validation text of "Email" field display when Forgot Password with "Email" field empty
 - ✓ Verify that validation text of "Email" field display when Forgot Password with "Email" field invalid email format and less than 6 characters.
 - ✓ Verify that validation text of "Email" field display when Forgot Password with "Email" field invalid email format and greater than 6 characters.
 - ✓ Verify that the forgot password can be cancel using the "Cancel" button
- ✓ Verify displays the password
 - ✓ Verify that the password can be displayed in text format when clicking on the "Eye" icon.
- ✓ Verify refresh page
 - ✓ Verify entered data not showing when Refresh button is clicked

3. KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐÓNG GÓP

- **Nâng cao kiến thức về kiểm thử phần mềm:**
 - ✓ Qua thực tập tại *Công Ty TNHH ARI Technology Joint Stock Company*, tôi đã nắm bắt được quy trình kiểm thử phần mềm trong doanh nghiệp, cũng như các kỹ thuật testing phần mềm hiện đại và phổ biến nhất hiện nay.
- **Đóng góp vào quá trình kiểm thử phần mềm của nhóm:**
 - ✓ Tôi đã tham gia vào các hoạt động kiểm thử phần mềm cho các dự án tại ARI, giúp tối ưu quy trình kiểm thử và đảm bảo chất lượng sản phẩm đạt yêu cầu.

CHƯƠNG 4.

TỔNG KẾT

Trong quá trình thực tập tại ***ARI Technology Joint Stock Company***, bản thân em được Ban giám đốc, cũng như các anh chị đồng nghiệp đã tạo điều kiện để em có thể học hỏi, làm việc thực tế và tiếp xúc với nhiều công việc khác nhau, như là: Nghiên cứu tài liệu, thực hiện automation testing, tham gia làm việc với tác phong chuyên nghiệp với môi trường làm việc Quốc tế. Qua những công việc đó, bản thân em học hỏi được rất nhiều kinh nghiệm từ thực tế và áp dụng được phần nào kiến thức đã học để làm việc. Tuy nhiên, vì thời gian thực tập ngắn (trong vòng 1 tháng) nên kiến thức thu thập sau khi thực tập chỉ mang tính bao quát và còn nhiều hạn chế. Nhưng, em cũng đã trang bị đủ cho mình các hành trang cần có để tiếp tục theo đuổi lĩnh vực Tester này.

Sau khi thực tập tại ***ARI Technology Joint Stock Company***, em thấy việc kiểm thử cho từng thiết bị cũng như vai trò của Tester là “không thể thiếu” trong mỗi dự án nào. Công việc của một Tester nhằm mục đích duy nhất là đảm bảo chất lượng đầu ra của sản phẩm ổn định. Việc kiểm thử trải qua nhiều giai đoạn để sản phẩm có chất lượng đầu ra tốt nhất. Tester và Developer cần phải hoạt động song song, phối hợp một cách nhịp nhàng với nhau. Nếu Developer là người tạo ra sản phẩm thì Tester là người đảm bảo sản phẩm đó hoạt động một cách trơn tru nhất. Ngày càng nhiều sản phẩm có chức năng giống nhau cùng cạnh tranh trên cùng thị trường thì sản phẩm nào có độ ổn định cao, ít lỗi, hiệu năng tốt, thì sẽ được khách hàng lựa chọn sử dụng. Vậy nên vai trò của Tester là “hết sức quan trọng”.

Một lần nữa, em xin gửi lời cảm ơn đến Quý công ty đã giúp đỡ và tạo điều kiện để em có thể tham gia thực tập trong thời gian vừa rồi. Em xin cảm ơn các Thầy Cô giáo khoa Điện, điện tử & Công nghệ vật liệu đã tạo điều kiện cho em đi thực tập sớm và hoàn thành báo cáo đúng thời hạn.