

TRƯỜNG ĐẠI HỌC KINH TẾ  
**KHOA THỐNG KÊ – TIN HỌC**



**BÁO CÁO THỰC TẬP NGHỀ NGHIỆP**

**NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ  
CHUYÊN NGÀNH TIN HỌC QUẢN LÝ**

**KIỂM THỬ THỦ CÔNG VÀ TỰ ĐỘNG  
CHO DỰ ÁN BANKING**

Sinh viên thực hiện : **Đỗ Thị Thy Hương**

Lớp : **47K14**

Đơn vị thực tập : **VNPT-IT Khu vực 3**

Cán bộ hướng dẫn : **Đào Lê Tùng**

Giảng viên hướng dẫn : **Ths. Nguyễn Thành Thủy**

***Đà Nẵng, 7/2024***

## NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP

Họ và tên sinh viên: Đỗ Thị Thụy Hương Lớp: 47K14

Khoa Thống kê – Tin học, Trường Đại học Kinh tế, Đại học Đà Nẵng

Thực tập từ ngày: 20/05/2024 đến ngày: 02/08/2024

Tên đơn vị thực tập: Tung tâm VNPT - IT Khu vực 3

Địa chỉ: 344 Đường 2 tháng 9, Hòa Cường Bắc, Hải Châu, Đà Nẵng

Số điện thoại liên hệ: 0855 539 335

Họ tên cán bộ hướng dẫn: Đào Lê Tung

Sau quá trình thực tập của sinh viên tại đơn vị, chúng tôi có một số đánh giá như sau:

STT	Nội dung đánh giá	Rất không tốt	Không tốt	Bình thường	Tốt	Rất tốt
1	Về thái độ, ý thức, đạo đức và việc tuân thủ các quy định và văn hóa đơn vị thực tập	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	Kiến thức chuyên môn	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	Khả năng hòa nhập, thích nghi và tác phong nghề nghiệp	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	Trách nhiệm, sáng tạo trong công việc	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

(Anh/chị vui lòng đánh dấu X vào ô tương ứng với năng lực của sinh viên)

**Ý kiến nhận xét và đề xuất** (Nhằm nâng cao chất lượng đào tạo, Nhà trường rất mong muốn nhận thêm những ý kiến khác từ quý doanh nghiệp):

Có tính thời đại học cao, có những ý tưởng sáng tạo và mới mẻ. Tuy nhiên còn cần được thêm kỹ năng làm việc nhóm và làm việc dưới áp lực để phát triển trong môi trường làm việc thực tế.

Điểm: 9.5

  
Hoàng Quốc Thái

Đà Nẵng, ngày 25 tháng 7 năm 2024

Xác nhận của đơn vị thực tập

 **PHÓ GIÁM ĐỐC**

  
Phan Văn Thảo

## LỜI CẢM ƠN

Để hoàn thành báo cáo thực tập nghề nghiệp này, trước tiên em xin gửi đến các quý thầy, cô giảng viên trường Đại học Kinh Tế - Đại học Đà Nẵng lời cảm ơn chân thành nhất. Nhận được sự giúp đỡ, chỉ bảo tận tình của quý thầy cô trong trường đã giúp em hoàn thành được bài báo cáo thực tập nghề nghiệp với đề tài “*Kiểm thử thủ công và tự động cho dự án Banking*”.

Để có đủ kỹ năng thực tế để hoàn thiện được bài báo cáo này, em cũng xin chân thành cảm ơn Ban giám đốc và các anh chị cán bộ, nhân viên bộ phận Tester của Trung tâm VNPT-IT Khu vực 3 thuộc Công ty Công nghệ Thông tin VNPT, Tập đoàn Bưu chính Viễn thông Việt Nam đã tạo điều kiện thuận lợi cho em được tìm hiểu thực tiễn trong suốt quá trình thực tập tại công ty.

Đặc biệt, em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Nguyễn Thành Thủy – Giảng viên hướng dẫn; và anh Đào Lê Tùng của Trung tâm đã hướng dẫn, nhắc nhở và giúp đỡ em trong quá trình thực tập nghề nghiệp.

Vì kiến thức bản thân còn nhiều hạn chế, trong quá trình thực tập, hoàn thiện báo cáo này em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ thầy cô cũng như quý công ty.

Em xin chân thành cảm ơn!

## LỜI CAM ĐOAN

Em xin cam đoan đề tài: ***“Kiểm thử thủ công và tự động cho dự án Banking”*** là đề tài, thành quả dựa trên sự nỗ lực của bản thân trong suốt quá trình thực tập tại Trung tâm VNPT-IT Khu vực 3 dựa trên cơ sở nghiên cứu lý thuyết đã học và những khảo sát thực tiễn dưới sự hướng dẫn của giảng viên hướng dẫn thầy Nguyễn Thành Thủy và cán bộ hướng dẫn anh Đào Lê Tùng.

Đồng thời, em xin cam kết rằng tất cả các thông tin, kết quả và ý kiến được trình bày trong báo cáo là chân thực và dựa trên thực tế cũng như quá trình nghiên cứu. Tất cả số liệu và kết quả nghiệm thu hoàn toàn không có sự sao chép, biên soạn hay sử dụng bất kỳ nguồn tài liệu nghiên cứu nào của người khác.

Nếu có bất kỳ sự gian lận nào, em xin chịu hoàn toàn trách nhiệm về nội dung báo cáo của mình.

## MỤC LỤC

LỜI CẢM ƠN .....	ii
LỜI CAM ĐOAN .....	iii
MỤC LỤC .....	iv
DANH MỤC HÌNH ẢNH .....	vii
DANH MỤC BẢNG BIỂU .....	viii
DANH MỤC CÁC TỪ VIẾT TẮT .....	ix
LỜI MỞ ĐẦU .....	1
CHƯƠNG 1. TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP VÀ CƠ SỞ LÝ THUYẾT .....	2
1.1. Giới thiệu tổng quát về doanh nghiệp thực tập .....	2
1.1.1. Giới thiệu chung.....	2
1.1.2. Tầm nhìn và sứ mệnh.....	2
1.1.3. Lĩnh vực hoạt động .....	3
1.2. Tổng quan về vị trí việc làm.....	3
1.2.1. Khái niệm tester .....	3
1.2.2. Vai trò của tester trong dự án CNTT .....	3
1.2.3. Kiến thức và kỹ năng cần thiết của tester .....	4
1.2.4. Lộ trình nghề nghiệp.....	5
1.3. Cơ sở lý thuyết về kiểm thử phần mềm.....	6
1.3.1. Kiểm thử phần mềm.....	6
1.3.2. Test case .....	8
1.3.3. Bug .....	10
1.3.4. Element .....	11
1.3.5. Locator .....	12
CHƯƠNG 2. THỰC HIỆN KIỂM THỬ THỦ CÔNG.....	13

2.1. Tổng quan dự án.....	13
2.1.1. Phân quyền người dùng .....	13
2.1.2. Mô tả chức năng.....	13
2.2. Thiết kế Test Case .....	15
2.2.1. New Account.....	15
2.2.2. New Customer.....	15
2.2.3. Balance Enquiry .....	18
2.2.4. Customized Statement.....	19
2.2.5. Delete Account.....	20
2.2.6. Delete Customer.....	20
2.2.7. Deposit .....	21
2.2.8. Edit Account .....	22
2.2.9. Edit Customer .....	22
2.2.10. Fund Transfer .....	25
2.2.11. Login .....	26
2.2.12. Mini Statement.....	27
2.2.13. Change Password .....	27
2.2.14. Withdraw.....	28
2.2.15. API .....	29
2.3. Kết quả kiểm thử .....	30
2.3.1. Kết quả thực hiện kiểm thử thủ công.....	30
2.3.2. Quản lý bug.....	31
<b>CHƯƠNG 3. THỰC HIỆN KIỂM THỬ TỰ ĐỘNG.....</b>	<b>32</b>
3.1. Công cụ sử dụng.....	32
3.1.1. Selenium WebDriver.....	32

3.1.2. TestNG .....	33
3.1.3. RestAssured .....	33
3.2. Thiết kế Test Script .....	33
3.2.1. Phạm vi kiểm thử .....	33
3.2.2. Page Objects.....	34
3.2.3. Test Scripts.....	39
3.3. Kết quả kiểm thử .....	45
3.3.1. API .....	45
3.3.2. Ứng dụng web.....	46
<b>CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>47</b>
<b>CHƯƠNG 5. CHECK LIST CỦA BÁO CÁO.....</b>	<b>48</b>
<b>CHƯƠNG 6. PHỤ LỤC.....</b>	<b>49</b>

## DANH MỤC HÌNH ẢNH

<i>Hình 1.1. Logo VNPT-IT .....</i>	<i>2</i>
<i>Hình 1.2. Lộ trình nghề nghiệp của tester.....</i>	<i>5</i>
<i>Hình 3.1. Mô hình hoạt động của Selenium Web Driver .....</i>	<i>32</i>
<i>Hình 3.2. Kết quả kiểm thử tự động API .....</i>	<i>45</i>
<i>Hình 3.3. Chi tiết lỗi test case API_TC_07 .....</i>	<i>45</i>
<i>Hình 3.4. Chi tiết lỗi test case API_TC_09 .....</i>	<i>45</i>
<i>Hình 3.5. Kết quả kiểm thử tự động các chức năng Login, Withdraw, Change Password, Fund Transfer .....</i>	<i>46</i>
<i>Hình 3.6. Chi tiết lỗi test case WD_TC_01 .....</i>	<i>46</i>



## DANH MỤC BẢNG BIỂU

<i>Bảng 2.1. Bảng mô tả tổng quát từng chức năng</i>	14
<i>Bảng 2.2. Danh sách test cases chức năng New Account</i>	15
<i>Bảng 2.3. Danh sách test cases chức năng New Customer</i>	18
<i>Bảng 2.4. Danh sách test cases chức năng Balance Enquiry</i>	18
<i>Bảng 2.5. Danh sách test cases chức năng Customized Statement</i>	19
<i>Bảng 2.6. Danh sách test cases chức năng Delete Account</i>	20
<i>Bảng 2.7. Danh sách test cases chức năng Delete Customer</i>	20
<i>Bảng 2.8. Danh sách test cases chức năng Deposit</i>	21
<i>Bảng 2.9. Danh sách test cases chức năng Edit Account</i>	22
<i>Bảng 2.10. Danh sách test cases chức năng New Customer</i>	25
<i>Bảng 2.11. Danh sách test cases chức năng Fund Transfer</i>	26
<i>Bảng 2.12. Danh sách test cases chức năng Login</i>	26
<i>Bảng 2.13. Danh sách test cases chức năng Mini Statement</i>	27
<i>Bảng 2.14. Danh sách test cases chức năng Change Password</i>	28
<i>Bảng 2.15. Danh sách test cases chức năng Withdraw</i>	28
<i>Bảng 2.16. Danh sách test cases API</i>	30
<i>Bảng 2.17. Kết quả kiểm thử thủ công</i>	30
<i>Bảng 2.18. Bảng thống kê Bug dựa trên mức độ nghiêm trọng và chức năng</i>	31
<i>Bảng 2.19. Danh sách Bug</i>	31

## **DANH MỤC CÁC TỪ VIẾT TẮT**

<b>QA</b>	: Quality Assurance
<b>QC</b>	: Quality Control
<b>BA</b>	: Business Analysis
<b>UX</b>	: User Experience
<b>UI</b>	: User Interface
<b>CNTT</b>	: Công nghệ thông tin
<b>API</b>	: Application Programming Interface

# LỜI MỞ ĐẦU

## 1. Mục tiêu của đề tài

- Tìm hiểu và nghiên cứu về các kiến thức căn bản của kiểm thử phần mềm cũng như các công cụ hỗ trợ việc kiểm thử phần mềm.
- Nghiên cứu chi tiết và thực hành việc viết test case dựa trên tài liệu yêu cầu, các kỹ thuật viết test case.
- Thực hành kiểm thử (thủ công và tự động) dựa trên các test case đã viết, báo cáo kết quả kiểm thử và quản lý bug.

## 2. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: ứng dụng Banking của Guru99
- Phạm vi nghiên cứu: Phân tích trang web, Xây dựng kịch bản kiểm thử, Thực hiện kiểm thử thủ công và tự động

## 3. Kết cấu của đề tài

Đề tài được tổ chức gồm phần mở đầu, 3 chương nội dung và phần kết luận:

- Mở đầu
- **Chương 1:** Tổng quan về đơn vị thực tập và cơ sở lý thuyết
- **Chương 2:** Thực hiện kiểm thử thủ công
- **Chương 3:** Thực hiện kiểm thử tự động
- Kết luận và hướng phát triển

## CHƯƠNG 1. TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP VÀ CƠ SỞ LÝ THUYẾT

### 1.1. Giới thiệu tổng quát về doanh nghiệp thực tập

#### 1.1.1. Giới thiệu chung

Công ty Công nghệ thông tin VNPT (*Tên viết tắt: VNPT-IT*) được thành lập theo Quyết định số 39/QĐ-VNPT-HĐTV-NL ngày 01 tháng 03 năm 2018 của Chủ tịch Tập đoàn Bưu chính Viễn thông Việt Nam, trên cơ sở tổ chức lại các nhiệm vụ và nguồn lực công nghệ thông tin thuộc Tập đoàn.



*Hình 1.1. Logo VNPT-IT*

VNPT-IT hoạt động trong lĩnh vực nghiên cứu phát triển, tích hợp các sản phẩm dịch vụ công nghệ thông tin phục vụ nội bộ Tập đoàn Bưu chính Viễn thông Việt Nam (VNPT) và các khách hàng bên ngoài VNPT (bao gồm cả khách hàng Quốc tế).

#### 1.1.2. Tầm nhìn và sứ mệnh

##### ***Tầm nhìn***

Với khát vọng tiên phong trong chiến lược đầu tư, cung cấp giải pháp, sản phẩm, dịch vụ và phát triển bền vững trong lĩnh vực công nghệ thông tin, Công ty công nghệ hàng đầu trong lĩnh vực công nghệ thông tin của Việt Nam (VNPT-IT) sẽ phấn đấu phát triển theo định hướng nghiên cứu, gia công, thử nghiệm để cung cấp các sản phẩm đa dạng, đa ngành nghề có uy tín đạt chất lượng cao nhằm đáp ứng nhu cầu xã hội góp phần nâng cao chất lượng cuộc sống của người Việt và nâng tầm vị thế của người Việt trên thị trường Quốc tế.

##### ***Sứ mệnh***

Là Công ty công nghệ trọng điểm Quốc gia, Công ty Công nghệ thông tin VNPT-IT thành viên của Tập đoàn Bưu chính Viễn thông Việt Nam sẽ đóng một vai trò chủ lực trong việc triển khai VNPT 4.0, tham gia mạnh mẽ vào quá trình chuyển đổi nền kinh tế

số của Việt Nam, cung cấp giải pháp xây dựng Chính quyền số cũng như là nền kinh tế số. VNPT-IT cũng sẽ là đơn vị cung cấp dịch vụ, giải pháp để chuyển đổi doanh nghiệp Việt Nam thành doanh nghiệp số và cung cấp các nền tảng để phát triển công nghệ số trong tương lai. VNPT-IT chính là hoài bão, là khát vọng mà VNPT hướng tới trên con đường khai phá vùng đất CNTT.

Chúng tôi mang trên mình sứ mệnh cũng như trọng trách cung cấp các giải pháp, sản phẩm, dịch vụ, sản phẩm thông minh cho xã hội tại tất cả các lĩnh vực như đào tạo, quản lý, chăm sóc sức khỏe, ứng dụng và chuyển giao công nghệ có chất lượng cao với thương hiệu và danh tiếng đạt đẳng cấp khu vực và Quốc tế, góp phần vào sự nghiệp công nghiệp hóa, hiện đại hóa đất nước trong bối cảnh hội nhập kinh tế Thế giới.

*“Vì một xã hội văn minh, mình bạch góp phần xây dựng cuộc sống người Việt tốt đẹp hơn”*

### **1.1.3. Lĩnh vực hoạt động**

Tổ chức nghiên cứu, phát triển, sản xuất, kinh doanh các sản phẩm, dịch vụ CNTT để cung cấp cho nội bộ VNPT và khách hàng bên ngoài VNPT;

Đầu tư, phát triển, quản lý tài sản các hệ thống, nền tảng CNTT; vận hành khai thác hệ thống điều hành sản xuất kinh doanh, đảm bảo an toàn, bảo mật thông tin cho các sản phẩm và các dịch vụ CNTT của VNPT cung cấp cho khách hàng.

## **1.2. Tổng quan về vị trí việc làm**

### **1.2.1. Khái niệm tester**

Tester là người có nhiệm vụ thực hiện các công việc chính như kiểm tra các lỗi, đảm bảo chất lượng sản phẩm tại từng công đoạn của quá trình sản xuất.

### **1.2.2. Vai trò của tester trong dự án CNTT**

Vai trò của tester quan trọng ngang bằng với lập trình viên cũng như công việc kiểm thử phần mềm so với công việc phát triển phần mềm.

Tester là người kiểm tra chất lượng phần mềm, kiểm tra để phát hiện các lỗi đang tồn tại trước khi giao sản phẩm cho khách hàng, để đảm bảo lỗi phải được tìm ra trước khi giao phần mềm cho khách hàng.

Tester đôi khi được một số công ty gọi là QC – Quality Control.

### **1.2.3. Kiến thức và kỹ năng cần thiết của tester**

Để thực hiện tốt vai trò của mình, tester cần có những kiến thức và kỹ năng:

*Kiến thức về kiểm thử:* Hiểu về quy trình kiểm thử; các kỹ thuật, phương pháp và những công cụ hỗ trợ liên quan

*Kỹ năng phân tích yêu cầu:* Có khả năng phân tích để hiểu rõ các tài liệu yêu cầu, nhằm xác định và xây dựng các kịch bản kiểm thử phù hợp

*Kỹ năng lập kế hoạch:* Có khả năng lên kế hoạch, xác định mức độ quan trọng và ưu tiên của công việc, cũng như ước lượng khối lượng công việc

*Kỹ năng báo cáo, ghi chép:* Có khả năng ghi chép lại chính xác và chi tiết kết quả kiểm thử, có khả năng báo cáo các lỗi xảy ra với đội ngũ phát triển

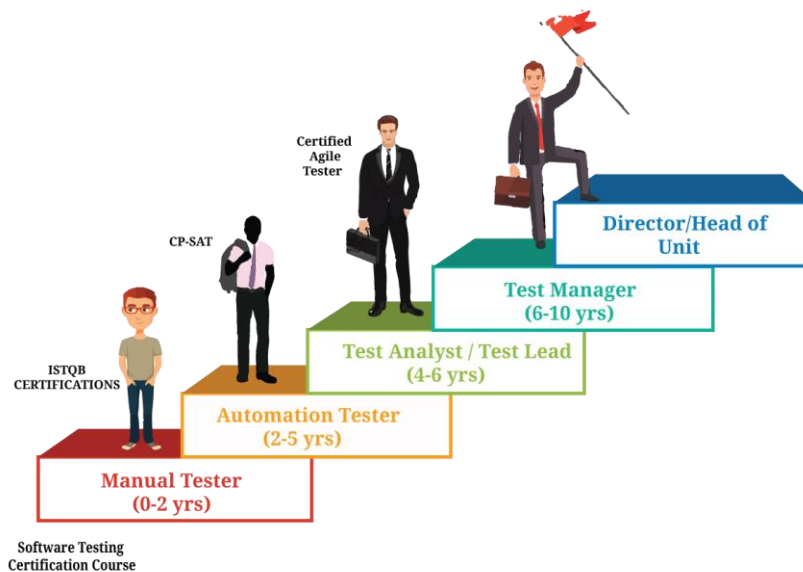
*Kỹ năng giao tiếp:* Có khả năng giao tiếp tốt với các thành viên của đội dự án, có khả năng diễn đạt và giải thích tốt

*Kỹ năng sử dụng các công cụ kiểm thử:* Có kiến thức và kỹ năng sử dụng các công cụ hỗ trợ và liên quan đến kiểm thử phần mềm phổ biến như Selenium, TestNG, Jira, ...

*Khả năng tiếp cận công nghệ, công cụ mới:* Có khả năng cập nhật kiến thức, công nghệ mới liên tục và kịp thời để tăng hiệu quả, năng suất kiểm thử và làm việc

*Kỹ năng lập trình (đối với Automation tester):* Có khả năng lập trình để tạo và thực thi các kịch bản kiểm thử tự động

#### 1.2.4. Lộ trình nghề nghiệp



Hình 1.2. Lộ trình nghề nghiệp của tester

Từ Lộ trình *nghề nghiệp của tester* có thể thấy, tester có một số cơ hội nghề nghiệp nhất định dựa trên mức độ kinh nghiệm cũng như chứng chỉ để chứng thực cho năng lực của tester:

*0-2 năm kinh nghiệm:* Kiểm thử viên thủ công (Manual Tester), thường yêu cầu hoặc ưu tiên những ứng viên có chứng chỉ ISTQB (*International Software Testing Qualifications Board*) ở mức độ Foundation level

*2-5 năm kinh nghiệm:* Có thể tiếp tục làm Kiểm thử viên thủ công hoặc bắt đầu với vị trí Kiểm thử viên tự động (Automation tester) khi đã có được kỹ năng cần thiết thông qua chứng chỉ CP-SAT (*Certified Professional – Selenium Automation Testing*) hoặc chứng chỉ ISTQB Specialist – Automation Software Tester

*4-6 năm kinh nghiệm:* Có thể tiếp tục làm Kiểm thử viên chuyên về nghiệp vụ hoặc bắt đầu các vị trí quản lý, quản trị như Leader

*6-10 năm kinh nghiệm:* Quản lý/Trưởng bộ phận/Trưởng nhóm Kiểm thử (Test Manager)

*Trên 10 năm kinh nghiệm:* Có cơ hội trở thành nhà quản trị, quản lý cấp cao của Công ty, doanh nghiệp, ...

## 1.3. Cơ sở lý thuyết về kiểm thử phần mềm

### 1.3.1. Kiểm thử phần mềm

#### a) Khái niệm kiểm thử phần mềm

Kiểm thử phần mềm là quá trình đánh giá và kiểm tra chất lượng, sự ổn định hay khả năng hoạt động đúng đắn của một phần mềm hay một phần, một chức năng nhỏ. Kiểm thử phần mềm là giai đoạn nhằm xác nhận khả năng sản phẩm có thể đáp ứng yêu cầu, kỳ vọng và mong đợi của khách hàng, người dùng hay không.

#### b) Các mục đích kiểm thử phần mềm

Về mục đích, kiểm thử phần mềm được chia làm 2 loại mục đích, tương ứng với 2 loại yêu cầu có thể có đối với một sản phẩm CNTT, trong đó bao gồm:

*Kiểm thử chức năng (Functional Testing):* Có trách nhiệm đảm bảo hệ thống có thể được thực thi đúng đắn và đáp ứng được các yêu cầu sử dụng

*Kiểm thử phi chức năng (Non-functional Testing):* Kiểm tra, đánh giá các yếu tố không ảnh hưởng trực tiếp đến khả năng vận hành của hệ thống như hiệu năng, UI, UX, bảo mật, tính tương thích, ...

#### c) Các giai đoạn kiểm thử phần mềm

##### *Kiểm thử đơn vị (Unit Testing)*

Kiểm thử đơn vị là giai đoạn kiểm thử trên những phần nhỏ và riêng lẻ của phần mềm. Mục đích của kiểm thử đơn vị là để đảm bảo mỗi chức năng nhỏ hay mỗi đoạn mã lệnh của phần mềm đều đang thực hiện đúng chức năng của nó và có thể đáp ứng được các yêu cầu, mong đợi. Kiểm thử đơn vị có thể được thực hiện bởi kiểm thử viên hoặc cũng có thể được thực hiện bởi các kỹ sư phần mềm ngay trong quá trình viết mã.

##### *Kiểm thử tích hợp (Integration Testing)*

Kiểm thử tích hợp là giai đoạn kiểm thử nhằm đảm bảo một nhóm các chức năng, một tập các phần nhỏ có liên quan nhau của phần mềm có thể đáp ứng được các yêu cầu, kỳ vọng của tất cả các chức năng hay các phần nhỏ đó. Việc kiểm thử này giúp đảm bảo các phần nhỏ của hệ thống đã được tích hợp mà không có tình trạng xung đột hay ảnh hưởng lẫn nhau.



### *Kiểm thử hệ thống (System Testing)*

Kiểm thử hệ thống là giai đoạn đánh giá, kiểm thử trên toàn bộ hệ thống sau khi tích hợp tất cả các chức năng, các phần nhỏ của phần mềm, từ đó thực hiện kiểm tra trên một phần mềm, hệ thống hoàn chỉnh.

### *Kiểm thử chấp nhận (Acceptance Testing)*

Kiểm thử chấp nhận là giai đoạn sản phẩm được đánh giá, kiểm tra lần cuối trước khi được bàn giao cho khách hàng hoặc phát hành chính thức cho người dùng. Mục đích của kiểm thử chấp nhận là để đảm bảo khách hàng hoặc người dùng cuối có thể sử dụng được hệ thống, phần mềm với đúng chức năng được thiết kế ban đầu.

Dựa trên kết quả mong muốn nhận được, có 3 kiểu kiểm thử chấp nhận phổ biến, bao gồm:

- *Kiểm thử chấp nhận người dùng (User Acceptance Testing – UAT)*: sản phẩm được thực hiện kiểm thử bởi chính người dùng hay khách hàng để đảm bảo sản phẩm đã đáp ứng được các yêu cầu hay mong đợi được đưa ra ban đầu
- *Kiểm thử Alpha*: sản phẩm được thực hiện kiểm thử bởi thành viên trong nội bộ công ty – ngoài *đội dự án* – để đảm bảo sản phẩm đã đáp ứng được các yêu cầu
- *Kiểm thử Beta*: sản phẩm được phát hành tạm thời đến một số người dùng giới hạn để được nhận những đánh giá, góp ý thực tế dựa trên trải nghiệm của chính họ, từ đó giúp *đội dự án* phát hiện thêm những vấn đề tồn đọng và có những sự cải thiện phù hợp trước khi phát hành chính thức và rộng rãi

#### d) Các kỹ thuật kiểm thử phần mềm

Tùy vào mục đích kiểm thử và từng giai đoạn kiểm thử mà kiểm thử viên có thể sử dụng các kỹ thuật, phương pháp khác nhau. Có một số kỹ thuật phổ biến như:

*Black box testing – Kiểm thử hộp đen*: Kỹ thuật này hướng đến việc kiểm tra phần mềm từ góc nhìn của người dùng cuối mà không biết và quan tâm đến cấu trúc bên trong của phần mềm, kỹ thuật này tập trung vào đầu vào và các kết quả thu được của đối tượng kiểm thử so với yêu cầu và mong đợi.

*White box testing – Kiểm thử hộp trắng:* Kỹ thuật này là kỹ thuật trái ngược với kiểm thử hộp đen khi tập trung vào kiểm thử cấu trúc nội bộ của phần mềm hay đối tượng kiểm thử, bao gồm cả những đoạn mã lệnh, dòng chảy dữ liệu, cấu trúc dữ liệu, ... do đó cần có sự hiểu biết nhất định đối với cấu trúc của hệ thống và cần được cấp quyền truy cập vào cơ sở dữ liệu.

*Gray box testing – Kiểm thử hộp xám:* Kỹ thuật này là một sự kết hợp của 2 kỹ thuật kiểm thử trên, người thực hiện kiểm thử hộp xám cũng cần hiểu biết về cấu trúc hệ thống và được phép truy cập vào cơ sở dữ liệu, đồng thời cũng cần có sự hiểu biết đối với yêu cầu của khách hàng hay người dùng cuối.

### **1.3.2. Test case**

#### **a) Khái niệm test case**

Test case là tập hợp các trường hợp điều kiện mà tester dựa vào đó để xác định ứng dụng, hệ thống phần mềm hoặc là 1 trong các chức năng của nó có hoạt động như mong muốn hay không. Test case được tester thiết kế dựa trên tài liệu đặc tả của BA cung cấp.

Test case được viết để đảm bảo phạm vi kiểm thử đã bao quát toàn bộ hệ thống, và để có thể tái sử dụng lại trong tương lai khi bất cứ ai cũng có thể tham khảo và tiến hành.

#### **b) Cấu trúc cơ bản của test case**

Cấu trúc cơ bản của một test case thường có 7 phần, bao gồm:

*Test case ID:* Đánh số ID theo thứ tự tăng dần, thường có chuỗi đại diện cho chức năng ở trước

*Tên test case:* Mô tả trường hợp sẽ kiểm thử, có thể viết ngắn gọn hoặc viết chi tiết như *user story*, tùy theo thói quen và văn hóa từng công ty

*Tiền điều kiện:* Điều kiện cần được thỏa mãn trước khi test case có thể được sử dụng trong kiểm thử

*Các bước thực hiện:* Mô tả các bước để thực hiện kiểm thử

*Data set:* Dữ liệu được sử dụng trong kiểm thử

*Kết quả mong đợi:* Mô tả kết quả mong đợi khi thực hiện các bước thực hiện trên

*Kết quả kiểm thử:* Là các kết quả Pass, Fail, ... ghi lại kết quả của việc kiểm thử

c) Quy trình thực hiện kiểm thử

Để việc kiểm thử được hiệu quả, tester cần tuân theo quy trình kiểm thử:

1. Cài đặt môi trường kiểm thử
2. Thực hiện kiểm thử theo bộ test case đã viết
3. Báo cáo lỗi (log bug) lên các công cụ quản lý lỗi
4. Theo dõi và thực hiện retest để verify các lỗi sau khi được chỉnh sửa và regression test trong trường hợp có sự thay đổi
5. Đóng lỗi và thực hiện cập nhật kết quả test

Sau khi kiểm thử, tester cũng cần thực hiện việc báo cáo kiểm thử, thông thường có những thông tin, số liệu như: số test case đã được thực hiện, số test case đã thông qua, số test case thất bại, số test case không thể thực hiện, kết luận mức độ đạt chỉ tiêu về chất lượng của chức năng so với chỉ tiêu chất lượng đã đề ra.

d) Một số kỹ thuật phát triển test case

*Phân vùng tương đương*

Phân vùng tương đương là một kỹ thuật được sử dụng trong kiểm thử hộp đen sử dụng việc chia nhỏ vùng dữ liệu đầu vào của một chương trình thành các vùng dữ liệu nhỏ, từ đó chọn một điểm dữ liệu đại diện cho mỗi vùng để thực hiện kiểm thử. Việc phân vùng thường được tuân theo sự thay đổi kết quả đầu ra giữa các vùng dữ liệu đầu vào.

*Giá trị biên*

Tương tự với phân vùng tương đương, kỹ thuật giá trị biên cũng chia nhỏ vùng dữ liệu đầu vào dựa trên sự thay đổi kết quả đầu ra. Tuy nhiên, kỹ thuật giá trị biên không lựa chọn dữ liệu bất kỳ để đại diện cho mỗi vùng mà thực hiện xác định các điểm giao nhau giữa các vùng dữ liệu đầu vào, từ đó chọn các điểm dữ liệu ngay trên, ngay dưới, và chính điểm giao giữa các vùng.

*Bảng quyết định*

Bảng quyết định cũng là một kỹ thuật được sử dụng trong kiểm thử hộp đen và thường được sử dụng khi kết quả đầu ra có sự phụ thuộc vào nhiều điều kiện đầu vào. Bảng quyết định giúp cung cấp một cái nhìn dạng bảng mô tả các trường hợp kết hợp khác nhau

của các điều kiện đầu vào dưới dạng Đúng – Sai (*True – False*). Từ đó, với mỗi sự kết hợp Đúng – Sai giữa các điều kiện sẽ cho ra một test case nhưng đôi khi người thực hiện kiểm thử cần loại bỏ những trường hợp không thể xảy ra để giảm lượng test case dư thừa.

#### *Đồ thị chuyển dịch trạng thái*

Kỹ thuật này cũng được sử dụng trong kiểm thử hộp đen khi người thực hiện sẽ dựa trên các trường hợp thay đổi trạng thái có thể xảy ra của một đối tượng để thực hiện viết test case, trong đó mỗi sự dịch chuyển tương ứng với một test case.

#### *Bao quát nhiều điều kiện*

Kỹ thuật này là một kỹ thuật trong kiểm thử hộp trắng, khi kết quả đầu ra phụ thuộc vào nhiều biến đầu vào mà mỗi biến lại có thể nhận được nhiều giá trị (*thường là dữ liệu rời rạc, không liên tục và cố định*). Với kỹ thuật này, để giảm bớt lượng công việc và số lượng test case mà test case sẽ được chọn dựa trên việc sắp xếp một cách có quy tắc các giá trị của từng biến sao cho với mỗi 2 biến bất kỳ, 2 giá trị bất kỳ từ 2 biến này luôn xuất hiện cùng nhau trong ít nhất 1 test case.

#### *Đoán lỗi*

Đoán lỗi là một kỹ thuật phụ thuộc vào kinh nghiệm của người thực hiện kiểm thử khi họ sử dụng kinh nghiệm của bản thân để dự đoán các lỗi có thể xảy ra trên hệ thống hay phần mềm để thực hiện việc kiểm thử.

#### *Khám phá*

Kỹ thuật kiểm thử khám phá cũng là một kỹ thuật dựa trên kinh nghiệm của kiểm thử viên khi họ không viết test case trước cho việc kiểm thử mà kiểm thử hệ thống theo thời gian thực. Kỹ thuật kiểm thử này chú trọng đến kiểm thử những hành vi người dùng nằm ngoài mong đợi của đội phát triển phần mềm.

### **1.3.3. Bug**

#### a) Khái niệm bug

Bug là những lỗi xuất hiện trong phần mềm hoặc hệ thống máy tính, dẫn đến những kết quả thực thi không chính xác hoặc phần mềm không hoạt động như mong muốn của người dùng. Bug có thể tồn tại trong toàn bộ phần mềm, từ chức năng cơ bản đến những tính năng phức tạp, hoặc cũng có thể là giao diện người dùng.

b) Cấu trúc cơ bản của bug

*Bug ID*: Đánh số ID theo thứ tự tăng dần.

*Tiêu đề (Title)*: Là tên của bug, thường là một mô tả ngắn gọn về bug.

*Các bước/quy trình tái hiện (Procedure/Steps)*: Là phần thân của bug report. Phần này giúp người đọc có thể tái hiện (*reproduce*) được bug. Khi viết quy trình nên đánh số thứ tự cho từng step để người đọc có thể dễ dàng tái hiện lại.

*Kết quả dự kiến (Expected Results)*: Mô tả kết quả mong đợi khi thực hiện các steps của phần *Procedure*.

*Kết quả thực tế (Actual Results)*: Mô tả kết quả thực tế nhận được khi thực hiện các steps của phần *Procedure*.

*Tệp đính kèm (Attachment)*: Cung cấp một hay nhiều tệp đính kèm bổ sung, thường là ảnh chụp màn hình/cảnh quay màn hình khi xuất hiện bug.

*Version web/app*: Cung cấp thông tin phiên bản của phần mềm, trình duyệt, ... để làm rõ môi trường xảy ra bug.

*Device/Account*: Cung cấp thông tin thiết bị, tài khoản được sử dụng khi xảy ra bug.

Ngoài ra, bug cũng có thể có thêm các thông tin khác tùy vào mục đích và nhu cầu của đội dự án hoặc doanh nghiệp.

#### **1.3.4. Element**

Element – hay Web Element – được sử dụng để chỉ một HTML element trên các trang web. Trong kiểm thử tự động, kiểm thử viên tương tác với các Element để thực hiện các tác vụ kiểm thử thông qua các câu lệnh, thường được các công cụ kiểm thử tự động nhận diện thông qua các *Locator*.

Element có thể là các *text box*, *button*, *form*, *table*, ...

### 1.3.5. *Locator*

Locator là các câu lệnh được sử dụng để cho các công cụ kiểm thử tự động nhận biết element nào trên trang web đang cần được tương tác. Các locator gồm:

*ID* – Đây là locator được phổ biến nhất khi sử dụng thuộc tính ID luôn luôn là duy nhất trên một trang web, cho phép người dùng có thể có thể dễ dàng định vị được element mình cần mà không gặp nhiều khó khăn.

*Name* – Đây cũng là một locator tương tự như *ID* nhưng sử dụng thuộc tính Name, cũng là một locator được sử dụng phổ biến.

*Class* – Trong nhiều trường hợp khi element cần tìm không có các thuộc tính ID hay Name, thuộc tính Class thường được sử dụng.

*Link Text* – Đây là một locator chỉ được sử dụng cho các element là các *hyperlink* trên trang web (thường là các thẻ `<a>` trong HTML), sử dụng phần văn bản hiển thị để thực hiện tìm kiếm.

*Partial Link Text* – Tương tự với *Link Text* nhưng chỉ thực hiện tìm kiếm dựa trên một phần văn bản hiển thị.

*Tag* – Đây là locator sử dụng tag name của các HTML element để thực hiện tìm kiếm trên trang web.

*CSS Selector* – Locator này sử dụng cách tìm kiếm tương tự như cách các HTML element được nhận diện trong thẻ `<style>` hay trong các tệp tin .css.

*Xpath* – Đây cũng là một locator được sử dụng phổ biến khi sử dụng xpath để nhận diện element nhờ sự linh hoạt và tiện dụng, tuy nhiên trong nhiều trường hợp xpath không được ưu tiên sử dụng do tốc độ chậm hơn các locator khác.

## CHƯƠNG 2. THỰC HIỆN KIỂM THỬ THỦ CÔNG

### 2.1. Tổng quan dự án

Guru99 Bank là một dự án Banking với mục đích tạo ra một ứng dụng ngân hàng online cho các khách hàng của mình.

#### 2.1.1. Phân quyền người dùng

Guru99 Bank cho phép người dùng đăng nhập với 2 roles là *Manager* và *Customer*.

##### a) Manager

Manager được sử dụng tất cả chức năng của hệ thống.

Các chức năng bao gồm: *New Customer, Edit Customer, Delete Customer, New Account, Edit Account, Delete Account, Deposit, Withdraw, Fund Transfer, Change Password, Balance Enquiry, Mini Statement, Customized Statement, Login & Logout*

##### b) Customer

Customer được sử dụng các chức năng: *Fund Transfer, Change Password, Balance Enquiry, Mini Statement, Customized Statement, Login & Logout*

#### 2.1.2. Mô tả chức năng

Tên chức năng	Roles	Mô tả
Balance Enquiry	Manager Customer	<b>Customer:</b> A customer can have multiple bank accounts. He can view balance of his accounts only <b>Manager:</b> A manager can view balance of all the customers who come under his supervision
Fund Transfer	Manager Customer	<b>Customer:</b> A customer can have transfer funds from his “own” account to any destination account. <b>Manager:</b> A manager can transfer funds from any source bank account to destination account
Mini Statement	Manager Customer	A Mini statement will show last 5 transactions of an account <b>Customer:</b> A customer can see mini-statement of only his “own” accounts <b>Manager:</b> A manager can see mini-statement of any account
Customized Statement	Manager Customer	A customized statement allows you to filter and display transactions in an account based on date, transaction value <b>Customer:</b> A customer can see Customized- statement of only his “own” accounts

		<b>Manager:</b> A manager can see Customized -statement of any account
Change Password	Manager Customer	<b>Customer:</b> A customer can change password of only his account. <b>Manager:</b> A manager can change password of only his account. He cannot change passwords of his customers
New Customer	Manager	<b>Manager:</b> A manager can add a new customer.
Edit Customer	Manager	<b>Manager:</b> A manager can edit details like address, email , telephone of a customer.
New Account	Manager	Currently system provides 2 types of accounts <ul style="list-style-type: none"> <li>• Saving</li> <li>• Current</li> </ul> A customer can have multiple saving accounts (one in his name , other in a joint name etc). He can have multiple current accounts for different companies he owns. Or he can have a multiple current and saving accounts. <b>Manager:</b> A manager can add a new account for an existing customer.
Edit Account	Manager	<b>Manager:</b> A manager can add a edit account details for an existing account
Delete Account	Manager	<b>Manager:</b> A manager can add a delete an account for a customer.
Delete Customer	Manager	A customer can be deleted only if he/she has no active current or saving accounts <b>Manager:</b> A manager can delete a customer.
Deposit	Manager	<b>Manager:</b> A manager can deposit money into any account. Usually done when cash is deposited at a bank branch.
Withdrawal	Manager	<b>Manager:</b> A manager can withdraw money from any account. Usually done when cash is withdrawn at a bank branch.

*Bảng 2.1. Bảng mô tả tổng quát từng chức năng*

Ngoài ra, Guru99 còn cung cấp một ứng dụng mobile cho phép người dùng thực hiện 2 chức năng *Mini Statement* và *Balance Enquiry*.

Bản đầy đủ tài liệu đặc tả hệ thống được dẫn link tại PHỤ LỤC.



## 2.2. Thiết kế Test Case

Ngoài các thông tin được liệt kê ngay sau, các thông tin đầy đủ của test case được viết trong tài liệu thiết kế test case được dẫn link tại PHỤ LỤC.

### 2.2.1. New Account

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
NA_TC_01	Không nhập Customer ID	1. Không nhập dữ liệu vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Customer ID is required" hiển thị
NA_TC_02	Nhập Customer ID chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	123!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
NA_TC_03	Nhập Customer ID chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	123Abcd	Thông báo lỗi "Characters are not allowed" hiển thị
NA_TC_04	Nhập Customer ID bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	123 14	Thông báo lỗi "First character cannot have space" hiển thị
NA_TC_05	Nhập Customer ID hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	76878	Không hiển thị thông báo lỗi
NA_TC_06	Không nhập Initial Deposit	1. Không nhập dữ liệu vào trường [Initial Deposit] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Initial Deposit must not be blank" hiển thị
NA_TC_07	Nhập Initial Deposit nhỏ hơn 500	1. Nhập dữ liệu nhỏ hơn 500 vào trường [Initial Deposit] 2. Nhấn Tab để đến trường tiếp theo	499	Hiện thị thông báo lỗi
NA_TC_08	Nhập Initial Deposit hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Initial Deposit] 2. Nhấn Tab để đến trường tiếp theo	500	Không hiển thị thông báo lỗi

Bảng 2.2. Danh sách test cases chức năng New Account

### 2.2.2. New Customer

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
NC_TC_01	Không nhập Customer Name	1. Không nhập dữ liệu vào trường [Customer Name] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Customer name must not be blank" hiển thị

NC_TC_02	Nhập Customer Name chứa ký tự số	1. Nhập dữ liệu chứa ký tự số vào trường [Customer Name] 2. Nhấn Tab để đến trường tiếp theo	Karen12	Thông báo lỗi "Numbers are not allowed" hiển thị
NC_TC_03	Nhập Customer Name chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Customer Name] 2. Nhấn Tab để đến trường tiếp theo	Karen!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
NC_TC_04	Nhập Customer Name bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Customer Name] 2. Nhấn Tab để đến trường tiếp theo	Karen	Thông báo lỗi "First character cannot have space" hiển thị
NC_TC_05	Nhập Customer Name hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Customer Name] 2. Nhấn Tab để đến trường tiếp theo	Karen	Không hiển thị thông báo lỗi
NC_TC_06	Không nhập Address	1. Không nhập dữ liệu vào trường [Address] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Address Field must not be blank" hiển thị
NC_TC_07	Nhập Address chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Address] 2. Nhấn Tab để đến trường tiếp theo	123!@# Pasteur	Thông báo lỗi "Special characters are not allowed" hiển thị
NC_TC_08	Nhập Address bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Address] 2. Nhấn Tab để đến trường tiếp theo	123 Pasteur	Thông báo lỗi "First character cannot have space" hiển thị
NC_TC_09	Nhập Address hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Address] 2. Nhấn Tab để đến trường tiếp theo	123 Pasteur	Không hiển thị thông báo lỗi
NC_TC_10	Không nhập City	1. Không nhập dữ liệu vào trường [City] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "City Field must not be blank" hiển thị
NC_TC_11	Nhập City chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [City] 2. Nhấn Tab để đến trường tiếp theo	New York City !@#	Thông báo lỗi "Special characters are not allowed" hiển thị
NC_TC_12	Nhập City chứa ký tự số	1. Nhập dữ liệu chứa ký tự số vào trường [City] 2. Nhấn Tab để đến trường tiếp theo	123New York City	Thông báo lỗi "Numbers are not allowed" hiển thị
NC_TC_13	Nhập City bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [City] 2. Nhấn Tab để đến trường tiếp theo	New York City	Thông báo lỗi "First character cannot have space" hiển thị
NC_TC_14	Nhập City hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [City] 2. Nhấn Tab để đến trường tiếp theo	New York City	Không hiển thị thông báo lỗi

NC_TC_15	Không nhập State	1. Không nhập dữ liệu vào trường [State] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "State must not be blank" hiển thị
NC_TC_16	Nhập State chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [State] 2. Nhấn Tab để đến trường tiếp theo	New York!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
NC_TC_17	Nhập State chứa ký tự số	1. Nhập dữ liệu chứa ký tự số vào trường [State] 2. Nhấn Tab để đến trường tiếp theo	123New York	Thông báo lỗi "Numbers are not allowed" hiển thị
NC_TC_18	Nhập State bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [State] 2. Nhấn Tab để đến trường tiếp theo	New York	Thông báo lỗi "First character cannot have space" hiển thị
NC_TC_19	Nhập State hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [State] 2. Nhấn Tab để đến trường tiếp theo	New York	Không hiển thị thông báo lỗi
NC_TC_20	Không nhập PIN	1. Không nhập dữ liệu vào trường [PIN] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "PIN Code must not be blank" hiển thị
NC_TC_21	Nhập PIN chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [PIN] 2. Nhấn Tab để đến trường tiếp theo	123!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
NC_TC_22	Nhập PIN chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [PIN] 2. Nhấn Tab để đến trường tiếp theo	123Abc	Thông báo lỗi "Numbers are not allowed" hiển thị
NC_TC_23	Nhập PIN bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [PIN] 2. Nhấn Tab để đến trường tiếp theo	12345	Thông báo lỗi "First character cannot have space" hiển thị
NC_TC_24	Nhập PIN có nhiều hơn 6 ký tự	1. Nhập dữ liệu nhiều hơn 6 ký tự vào trường [PIN] 2. Nhấn Tab để đến trường tiếp theo	1234567	Thông báo lỗi "PIN Code must have 6 Digits" hiển thị
NC_TC_25	Nhập PIN có ít hơn 6 ký tự	1. Nhập dữ liệu ít hơn 6 ký tự vào trường [PIN] 2. Nhấn Tab để đến trường tiếp theo	1234	Thông báo lỗi "PIN Code must have 6 Digits" hiển thị
NC_TC_26	Nhập PIN hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [PIN] 2. Nhấn Tab để đến trường tiếp theo	123456	Không hiển thị thông báo lỗi
NC_TC_27	Không nhập Mobile Number	1. Không nhập dữ liệu vào trường [Mobile Number] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Mobile no must not be blank" hiển thị
NC_TC_28	Nhập Mobile Number chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Mobile Number] 2. Nhấn Tab để đến trường tiếp theo	987456!@#	Thông báo lỗi "Special characters are not allowed" hiển thị

NC_TC_29	Nhập Mobile Number chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Mobile Number] 2. Nhấn Tab để đến trường tiếp theo	987456Abc	Thông báo lỗi "Numbers are not allowed" hiển thị
NC_TC_30	Nhập Mobile Number bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Mobile Number] 2. Nhấn Tab để đến trường tiếp theo	987456	Thông báo lỗi "First character cannot have space" hiển thị
NC_TC_31	Nhập Mobile hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Mobile Number] 2. Nhấn Tab để đến trường tiếp theo	987456	Không hiển thị thông báo lỗi
NC_TC_32	Không nhập Email	1. Không nhập dữ liệu vào trường [Email] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Email ID must not be blank" hiển thị
NC_TC_33	Nhập Email không đúng định dạng	1. Nhập dữ liệu không đúng định dạng ***@***.*** vào trường [Email] 2. Nhấn Tab để đến trường tiếp theo	karen karen@gmail karen@gmail.com karen@ karen@gmail.	Thông báo lỗi "Email ID is not valid" hiển thị
NC_TC_34	Nhập Email bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Email] 2. Nhấn Tab để đến trường tiếp theo	' karen@gmail.com'	Thông báo lỗi "First character cannot have space" hiển thị
NC_TC_35	Nhập Email hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Email] 2. Nhấn Tab để đến trường tiếp theo	karen@gmail.com	Không hiển thị thông báo lỗi

*Bảng 2.3. Danh sách test cases chức năng New Customer*

### 2.2.3. Balance Enquiry

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
BE_TC_01	Không nhập Account No	1. Không nhập dữ liệu vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Account No must not be blank" hiển thị
BE_TC_02	Nhập Account No chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1245!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
BE_TC_03	Nhập Account No chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1345Abc	Thông báo lỗi "Characters are not allowed" hiển thị
BE_TC_04	Nhập Account No hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	136235	Không hiển thị thông báo lỗi

*Bảng 2.4. Danh sách test cases chức năng Balance Enquiry*

#### 2.2.4. Customized Statement

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
CS_TC_01	Không nhập Account No	1. Không nhập dữ liệu vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Account No must not be blank" hiển thị
CS_TC_02	Nhập Account No chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1245!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
CS_TC_03	Nhập Account No chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1345Abc	Thông báo lỗi "Characters are not allowed" hiển thị
CS_TC_04	Nhập Account No hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	136235	Không hiển thị thông báo lỗi
CS_TC_05	Không nhập Minimum Transaction Value	1. Không nhập dữ liệu vào trường [Minimum Transaction Value] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Amount Lower Limit is required" hiển thị
CS_TC_06	Nhập Minimum Transaction Value chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Minimum Transaction Value] 2. Nhấn Tab để đến trường tiếp theo	100@#\$	Thông báo lỗi "Special characters are not allowed" hiển thị
CS_TC_07	Nhập Minimum Transaction Value chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Minimum Transaction Value] 2. Nhấn Tab để đến trường tiếp theo	100avd	Thông báo lỗi "Characters are not allowed" hiển thị
CS_TC_08	Nhập Minimum Transaction Value hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Minimum Transaction Value] 2. Nhấn Tab để đến trường tiếp theo	100	Không hiển thị thông báo lỗi
CS_TC_09	Không nhập Number of Transaction	1. Không nhập dữ liệu vào trường [Number of Transaction] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Minimum Transaction Value must not be blank" hiển thị
CS_TC_10	Nhập Number of Transaction chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Number of Transaction] 2. Nhấn Tab để đến trường tiếp theo	25#	Thông báo lỗi "Special characters are not allowed" hiển thị
CS_TC_11	Nhập Number of Transaction chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Number of Transaction] 2. Nhấn Tab để đến trường tiếp theo	25f	Thông báo lỗi "Characters are not allowed" hiển thị
CS_TC_12	Nhập Number of Transaction hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Number of Transaction] 2. Nhấn Tab để đến trường tiếp theo	25	Không hiển thị thông báo lỗi

Bảng 2.5. Danh sách test cases chức năng Customized Statement

### 2.2.5. Delete Account

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
DA_TC_01	Không nhập Account No	1. Không nhập dữ liệu vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Account No must not be blank" hiển thị
DA_TC_02	Nhập Account No chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1245!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
DA_TC_03	Nhập Account No chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1345Abc	Thông báo lỗi "Characters are not allowed" hiển thị
DA_TC_04	Nhập Account No hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	136235	Không hiển thị thông báo lỗi

Bảng 2.6. Danh sách test cases chức năng Delete Account

### 2.2.6. Delete Customer

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
DC_TC_01	Không nhập Customer ID	1. Không nhập dữ liệu vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Customer ID is required" hiển thị
DC_TC_02	Nhập Customer ID chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	123!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
DC_TC_03	Nhập Customer ID chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	123Abcd	Thông báo lỗi "Characters are not allowed" hiển thị
DC_TC_04	Nhập Customer ID bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	123 14	Thông báo lỗi "First character cannot have space" hiển thị
DC_TC_05	Nhập Customer ID hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	76878	Không hiển thị thông báo lỗi

Bảng 2.7. Danh sách test cases chức năng Delete Customer

### 2.2.7. Deposit

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
DE_TC_01	Không nhập Account No	1. Không nhập dữ liệu vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Account No must not be blank" hiển thị
DE_TC_02	Nhập Account No chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1245!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
DE_TC_03	Nhập Account No chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1345Abc	Thông báo lỗi "Characters are not allowed" hiển thị
DE_TC_04	Nhập Account No hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	136235	Không hiển thị thông báo lỗi
DE_TC_05	Không nhập Amount	1. Không nhập dữ liệu vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Amount field must not be blank" hiển thị
DE_TC_06	Nhập Amount chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo	200%\$	Thông báo lỗi "Special characters are not allowed" hiển thị
DE_TC_07	Nhập Amount chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo	200abc	Thông báo lỗi "Characters are not allowed" hiển thị
DE_TC_08	Nhập Amount hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo	200	Không hiển thị thông báo lỗi
DE_TC_09	Không nhập Description	1. Không nhập dữ liệu vào trường [Description] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Description must not be blank" hiển thị
DE_TC_10	Nhập Description hợp lệ	1. Nhập dữ liệu bất kỳ vào trường [Description] 2. Nhấn Tab để đến trường tiếp theo		Không hiển thị thông báo lỗi

Bảng 2.8. Danh sách test cases chức năng Deposit

### 2.2.8. Edit Account

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
EA_TC_01	Không nhập Account No	1. Không nhập dữ liệu vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Account No must not be blank" hiển thị
EA_TC_02	Nhập Account No chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1245!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
EA_TC_03	Nhập Account No chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1345Abc	Thông báo lỗi "Characters are not allowed" hiển thị
EA_TC_04	Nhập Account No hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	136235	Không hiển thị thông báo lỗi

Bảng 2.9. Danh sách test cases chức năng Edit Account

### 2.2.9. Edit Customer

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
EC_TC_01	Không nhập Customer ID	1. Không nhập dữ liệu vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Customer ID is required" hiển thị
EC_TC_02	Nhập Customer ID chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	123!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
EC_TC_03	Nhập Customer ID chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	123Abcd	Thông báo lỗi "Characters are not allowed" hiển thị
EC_TC_04	Nhập Customer ID bắt đầu bằng ký tự trắng	1. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	123 14	Thông báo lỗi "First character cannot have space" hiển thị
EC_TC_05	Nhập Customer ID hợp lệ	1. Nhập dữ liệu hợp lệ vào [Customer ID] 2. Nhấn Tab để đến trường tiếp theo	76878	Không hiển thị thông báo lỗi
EC_TC_06	Không nhập Address	1. Xóa dữ liệu cũ của trường [Address] 2. Không nhập dữ liệu vào trường [Address] 3. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Address Field must not be blank" hiển thị



EC_TC_0 7	Nhập Address chứa ký tự đặc biệt	1. Xóa dữ liệu cũ của trường [Address] 2. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Address] 3. Nhấn Tab để đến trường tiếp theo	123!@# Pasteur	Thông báo lỗi "Special characters are not allowed" hiển thị
EC_TC_0 8	Nhập Address bắt đầu bằng ký tự trắng	1. Xóa dữ liệu cũ của trường [Address] 2. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Address] 3. Nhấn Tab để đến trường tiếp theo	123 Pasteur	Thông báo lỗi "First character cannot have space" hiển thị
EC_TC_0 9	Nhập Address hợp lệ	1. Xóa dữ liệu cũ của trường [Address] 2. Nhập dữ liệu hợp lệ vào trường [Address] 3. Nhấn Tab để đến trường tiếp theo	123 Pasteur	Không hiển thị thông báo lỗi
EC_TC_1 0	Không nhập City	1. Xóa dữ liệu cũ của trường [City] 2. Không nhập dữ liệu vào trường [City] 3. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "City Field must not be blank" hiển thị
EC_TC_1 1	Nhập City chứa ký tự đặc biệt	1. Xóa dữ liệu cũ của trường [City] 2. Nhập dữ liệu chứa ký tự đặc biệt vào trường [City] 3. Nhấn Tab để đến trường tiếp theo	New York City !@#	Thông báo lỗi "Special characters are not allowed" hiển thị
EC_TC_1 2	Nhập City chứa ký tự số	1. Xóa dữ liệu cũ của trường [City] 2. Nhập dữ liệu chứa ký tự số vào trường [City] 3. Nhấn Tab để đến trường tiếp theo	123New York City	Thông báo lỗi "Numbers are not allowed" hiển thị
EC_TC_1 3	Nhập City bắt đầu bằng ký tự trắng	1. Xóa dữ liệu cũ của trường [City] 2. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [City] 3. Nhấn Tab để đến trường tiếp theo	New York City	Thông báo lỗi "First character cannot have space" hiển thị
EC_TC_1 4	Nhập City hợp lệ	1. Xóa dữ liệu cũ của trường [City] 2. Nhập dữ liệu hợp lệ vào trường [City] 3. Nhấn Tab để đến trường tiếp theo	New York City	Không hiển thị thông báo lỗi
EC_TC_1 5	Không nhập State	1. Xóa dữ liệu cũ của trường [State] 2. Không nhập dữ liệu vào trường [State] 3. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "State must not be blank" hiển thị
EC_TC_1 6	Nhập State chứa ký tự đặc biệt	1. Xóa dữ liệu cũ của trường [State] 2. Nhập dữ liệu chứa ký tự đặc biệt vào trường [State] 3. Nhấn Tab để đến trường tiếp theo	New York!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
EC_TC_1 7	Nhập State chứa ký tự số	1. Xóa dữ liệu cũ của trường [State] 2. Nhập dữ liệu chứa ký tự số vào trường [State] 3. Nhấn Tab để đến trường tiếp theo	123New York	Thông báo lỗi "Numbers are not allowed" hiển thị
EC_TC_1 8	Nhập State bắt đầu bằng ký tự trắng	1. Xóa dữ liệu cũ của trường [State] 2. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [State] 3. Nhấn Tab để đến trường tiếp theo	New York	Thông báo lỗi "First character cannot have space" hiển thị

EC_TC_1 9	Nhập State hợp lệ	1. Xóa dữ liệu cũ của trường [State] 2. Nhập dữ liệu hợp lệ vào trường [State] 3. Nhấn Tab để đến trường tiếp theo	New York	Không hiển thị thông báo lỗi
EC_TC_2 0	Không nhập PIN	1. Xóa dữ liệu cũ của trường [PIN] 2. Không nhập dữ liệu vào trường [PIN] 3. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "PIN Code must not be blank" hiển thị
EC_TC_2 1	Nhập PIN chứa ký tự đặc biệt	1. Xóa dữ liệu cũ của trường [PIN] 2. Nhập dữ liệu chứa ký tự đặc biệt vào trường [PIN] 3. Nhấn Tab để đến trường tiếp theo	123!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
EC_TC_2 2	Nhập PIN chứa ký tự chữ	1. Xóa dữ liệu cũ của trường [PIN] 2. Nhập dữ liệu chứa ký tự chữ vào trường [PIN] 3. Nhấn Tab để đến trường tiếp theo	123Abc	Thông báo lỗi "Numbers are not allowed" hiển thị
EC_TC_2 3	Nhập PIN bắt đầu bằng ký tự trắng	1. Xóa dữ liệu cũ của trường [PIN] 2. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [PIN] 3. Nhấn Tab để đến trường tiếp theo	12345	Thông báo lỗi "First character cannot have space" hiển thị
EC_TC_2 4	Nhập PIN có nhiều hơn 6 ký tự	1. Xóa dữ liệu cũ của trường [PIN] 2. Nhập dữ liệu nhiều hơn 6 ký tự vào trường [PIN] 3. Nhấn Tab để đến trường tiếp theo	1234567	Thông báo lỗi "PIN Code must have 6 Digits" hiển thị
EC_TC_2 5	Nhập PIN có ít hơn 6 ký tự	1. Xóa dữ liệu cũ của trường [PIN] 2. Nhập dữ liệu ít hơn 6 ký tự vào trường [PIN] 3. Nhấn Tab để đến trường tiếp theo	1234	Thông báo lỗi "PIN Code must have 6 Digits" hiển thị
EC_TC_2 6	Nhập PIN hợp lệ	1. Xóa dữ liệu cũ của trường [PIN] 2. Nhập dữ liệu hợp lệ vào trường [PIN] 3. Nhấn Tab để đến trường tiếp theo	123456	Không hiển thị thông báo lỗi
EC_TC_2 7	Không nhập Mobile Number	1. Xóa dữ liệu cũ của trường [Mobile Number] 2. Không nhập dữ liệu vào trường [Mobile Number] 3. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Mobile no must not be blank" hiển thị
EC_TC_2 8	Nhập Mobile Number chứa ký tự đặc biệt	1. Xóa dữ liệu cũ của trường [Mobile Number] 2. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Mobile Number] 3. Nhấn Tab để đến trường tiếp theo	987456!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
EC_TC_2 9	Nhập Mobile Number chứa ký tự chữ	1. Xóa dữ liệu cũ của trường [Mobile Number] 2. Nhập dữ liệu chứa ký tự chữ vào trường [Mobile Number] 3. Nhấn Tab để đến trường tiếp theo	987456Abc	Thông báo lỗi "Numbers are not allowed" hiển thị
EC_TC_3 0	Nhập Mobile Number bắt đầu bằng ký tự trắng	1. Xóa dữ liệu cũ của trường [Mobile Number] 2. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Mobile Number] 3. Nhấn Tab để đến trường tiếp theo	987456	Thông báo lỗi "First character cannot have space" hiển thị

EC_TC_3 1	Nhập Mobile hợp lệ	1. Xóa dữ liệu cũ của trường [Mobile Number] 2. Nhập dữ liệu hợp lệ vào trường [Mobile Number] 3. Nhấn Tab để đến trường tiếp theo	987456	Không hiển thị thông báo lỗi
EC_TC_3 2	Không nhập Email	1. Xóa dữ liệu cũ của trường [Email] 2. Không nhập dữ liệu vào trường [Email] 3. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Email ID must not be blank" hiển thị
EC_TC_3 3	Nhập Email không đúng định dạng	1. Xóa dữ liệu cũ của trường [Email] 2. Nhập dữ liệu không đúng định dạng ***@***.*** vào trường [Email] 3. Nhấn Tab để đến trường tiếp theo	karen karen@gmail karen.gmail.com karen@ karen@gmail.	Thông báo lỗi "Email ID is not valid" hiển thị
EC_TC_3 4	Nhập Email bắt đầu bằng ký tự trắng	1. Xóa dữ liệu cũ của trường [Email] 2. Nhập dữ liệu chứa ký tự trắng ở đầu vào trường [Email] 3. Nhấn Tab để đến trường tiếp theo	' karen@gmail.co m'	Thông báo lỗi "First character cannot have space" hiển thị
EC_TC_3 5	Nhập Email hợp lệ	1. Xóa dữ liệu cũ của trường [Email] 2. Nhập dữ liệu hợp lệ vào trường [Email] 3. Nhấn Tab để đến trường tiếp theo	karen@gmail.co m	Không hiển thị thông báo lỗi

*Bảng 2.10. Danh sách test cases chức năng New Customer*

### **2.2.10.Fund Transfer**

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
FT_TC_01	Không nhập Payers Account Number	1. Không nhập dữ liệu vào trường [Payers Account Number] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Payers Account Number must not be blank" hiển thị
FT_TC_02	Nhập Payers Account Number chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Payers Account Number] 2. Nhấn Tab để đến trường tiếp theo	1245!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
FT_TC_03	Nhập Payers Account Number chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Payers Account Number] 2. Nhấn Tab để đến trường tiếp theo	1345Abc	Thông báo lỗi "Characters are not allowed" hiển thị
FT_TC_04	Nhập Payers Account Number hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Payers Account Number] 2. Nhấn Tab để đến trường tiếp theo	136236	Không hiển thị thông báo lỗi
FT_TC_05	Không nhập Payees Account Number	1. Không nhập dữ liệu vào trường [Payees Account Number] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Payees Account Number must not be blank" hiển thị
FT_TC_06	Nhập Payees Account Number chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Payees Account Number] 2. Nhấn Tab để đến trường tiếp theo	1245!@#	Thông báo lỗi "Special characters are not allowed" hiển thị

FT_TC_07	Nhập Payees Account Number chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Payees Account Number] 2. Nhấn Tab để đến trường tiếp theo	1345Abc	Thông báo lỗi "Characters are not allowed" hiển thị
FT_TC_08	Nhập Payees Account Number hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Payees Account Number] 2. Nhấn Tab để đến trường tiếp theo	136235	Không hiển thị thông báo lỗi
FT_TC_09	Không nhập Amount	1. Không nhập dữ liệu vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Amount Field must not be blank" hiển thị
FT_TC_10	Nhập Amount chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo	200%\$	Thông báo lỗi "Special characters are not allowed" hiển thị
FT_TC_11	Nhập Amount chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo	200abc	Thông báo lỗi "Characters are not allowed" hiển thị
FT_TC_12	Nhập Amount hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo	200	Không hiển thị thông báo lỗi
FT_TC_13	Không nhập Description	1. Không nhập dữ liệu vào trường [Description] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Description cannot be blank" hiển thị
FT_TC_14	Nhập Description hợp lệ	1. Nhập dữ liệu bất kỳ vào trường [Description] 2. Nhấn Tab để đến trường tiếp theo		Không hiển thị thông báo lỗi

*Bảng 2.11. Danh sách test cases chức năng Fund Transfer*

### 2.2.11.Login

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
LI_TC_01	Không nhập User ID	1. Không nhập dữ liệu vào trường [User ID] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "User-ID must not be blank" hiển thị
LI_TC_02	Nhập User ID hợp lệ	1. Nhập dữ liệu bất kỳ vào trường [User ID] 2. Nhấn Tab để đến trường tiếp theo		Không hiển thị thông báo lỗi
LI_TC_03	Không nhập Password	1. Không nhập dữ liệu vào trường [Password] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Password must not be blank" hiển thị
LI_TC_04	Nhập Password hợp lệ	1. Nhập dữ liệu bất kỳ vào trường [Password] 2. Nhấn Tab để đến trường tiếp theo		Không hiển thị thông báo lỗi

*Bảng 2.12. Danh sách test cases chức năng Login*

### 2.2.12. Mini Statement

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
MS_TC_01	Không nhập Account No	1. Không nhập dữ liệu vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Account No must not be blank" hiển thị
MS_TC_02	Nhập Account No chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1245!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
MS_TC_03	Nhập Account No chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1345Abc	Thông báo lỗi "Characters are not allowed" hiển thị
MS_TC_04	Nhập Account No hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	136235	Không hiển thị thông báo lỗi

Bảng 2.13. Danh sách test cases chức năng Mini Statement

### 2.2.13. Change Password

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
CP_TC_01	Không nhập Old Password	1. Không nhập dữ liệu vào trường [Old Password] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Old Password must not be blank" hiển thị
CP_TC_02	Nhập Old Password hợp lệ	1. Nhập dữ liệu bất kỳ vào trường [Old Password] 2. Nhấn Tab để đến trường tiếp theo	123456abc	Không hiển thị thông báo lỗi
CP_TC_03	Không nhập New Password	1. Không nhập dữ liệu vào trường [New Password] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "New Password must not be blank" hiển thị
CP_TC_04	Nhập New Password không chứa ký tự số	1. Nhập dữ liệu không chứa ký tự số vào trường [New Password] 2. Nhấn Tab để đến trường tiếp theo	abcd!@#\$	Thông báo lỗi "Enter at-least one numeric value" hiển thị
CP_TC_05	Nhập New Password không chứa ký tự đặc biệt	1. Nhập dữ liệu không chứa ký tự đặc biệt vào trường [New Password] 2. Nhấn Tab để đến trường tiếp theo	abcd1234	Thông báo lỗi "Enter at-least one special character" hiển thị
CP_TC_06	Nhập New Password hợp lệ	1. Nhập dữ liệu chứa cả ký tự số và ký tự đặc biệt vào trường [New Password] 2. Nhấn Tab để đến trường tiếp theo	1234!@#\$	Không hiển thị thông báo lỗi
CP_TC_07	Không nhập Confirm Password	1. Không nhập dữ liệu vào trường [Confirm Password] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Confirm Password" must not be blank" hiển thị

CP_TC_08	Nhập Confirm Password không trùng với New Password	1. Nhập dữ liệu hợp lệ vào trường [New Password] 2. Nhập dữ liệu khác dữ liệu đã nhập vào trường [Confirm Password] 3. Nhấn Tab để đến trường tiếp theo	- New Pass: 1234!@#\$ - Confirm Pass: 1234	Thông báo lỗi "Passwords do not match" hiển thị
CP_TC_09	Nhập Confirm Password hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [New Password] 2. Nhập lại dữ liệu vào trường [Confirm Password] 3. Nhấn Tab để đến trường tiếp theo	- New Pass: 1234!@#\$ - Confirm Pass: 1234!@#\$	Không hiển thị thông báo lỗi

*Bảng 2.14. Danh sách test cases chức năng Change Password*

#### 2.2.14. Withdraw

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
WD_TC_01	Không nhập Account No	1. Không nhập dữ liệu vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Account No must not be blank" hiển thị
WD_TC_02	Nhập Account No chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1245!@#	Thông báo lỗi "Special characters are not allowed" hiển thị
WD_TC_03	Nhập Account No chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	1345Abc	Thông báo lỗi "Characters are not allowed" hiển thị
WD_TC_04	Nhập Account No hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Account No] 2. Nhấn Tab để đến trường tiếp theo	136235	Không hiển thị thông báo lỗi
WD_TC_05	Không nhập Amount	1. Không nhập dữ liệu vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Amount field must not be blank" hiển thị
WD_TC_06	Nhập Amount chứa ký tự đặc biệt	1. Nhập dữ liệu chứa ký tự đặc biệt vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo	200%\$	Thông báo lỗi "Special characters are not allowed" hiển thị
WD_TC_07	Nhập Amount chứa ký tự chữ	1. Nhập dữ liệu chứa ký tự chữ vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo	200abc	Thông báo lỗi "Characters are not allowed" hiển thị
WD_TC_08	Nhập Amount hợp lệ	1. Nhập dữ liệu hợp lệ vào trường [Amount] 2. Nhấn Tab để đến trường tiếp theo	200	Không hiển thị thông báo lỗi
WD_TC_09	Không nhập Description	1. Không nhập dữ liệu vào trường [Description] 2. Nhấn Tab để đến trường tiếp theo		Thông báo lỗi "Description must not be blank" hiển thị
WD_TC_10	Nhập Description hợp lệ	1. Nhập dữ liệu bất kỳ vào trường [Description] 2. Nhấn Tab để đến trường tiếp theo		Không hiển thị thông báo lỗi

*Bảng 2.15. Danh sách test cases chức năng Withdraw*

## 2.2.15.API

Testcase ID	Tên testcase mô tả	Các bước thực hiện	Dataset	Kết quả mong đợi
API_TC_01	Không nhập Account No khi sử dụng Mini Statement API	1. Không nhập Account No vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkministatement.php?CUSTOMER_ID=68195&PASSWORD=1234!&Account_No=	Thông báo lỗi "NoData" hiển thị dưới dạng JSON
API_TC_02	Nhập Account No không tồn tại trong hệ thống khi sử dụng Mini Statement API	1. Nhập Account No không tồn tại vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkministatement.php?CUSTOMER_ID=68195&PASSWORD=1234!&Account_No=145	Thông báo lỗi "NoData" hiển thị dưới dạng JSON
API_TC_03	Nhập Customer ID không tồn tại trong hệ thống khi sử dụng Mini Statement API	1. Nhập Customer ID không tồn tại vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkministatement.php?CUSTOMER_ID=681&PASSWORD=1234!&Account_No=1	Thông báo lỗi "Login Credentials Incorrect" hiển thị dưới dạng JSON
API_TC_04	Không nhập Password khi sử dụng Mini Statement API	1. Không nhập Password vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkministatement.php?CUSTOMER_ID=68195&PASSWORD=&Account_No=1	Thông báo lỗi "Login Credentials Incorrect" hiển thị dưới dạng JSON
API_TC_05	Nhập sai Password khi sử dụng Mini Statement API	1. Nhập Password sai vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkministatement.php?CUSTOMER_ID=68195&PASSWORD=123&Account_No=1	Thông báo lỗi "Login Credentials Incorrect" hiển thị dưới dạng JSON
API_TC_06	Nhập tất cả thông tin hợp lệ khi sử dụng Mini Statement API	1. Nhập Customer ID, Password, Account No hợp lệ vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkministatement.php?CUSTOMER_ID=68195&PASSWORD=1234!&Account_No=1	Thông báo thành công "Success" và kết quả Mini Statement được hiển thị dưới dạng JSON
API_TC_07	Không nhập Account No khi sử dụng Balance EnquiryAPI	1. Không nhập Account No vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkbalanceenquiry.php?CUSTOMER_ID=68195&PASSWORD=1234!&Account_No=	Thông báo thành công "Success" và kết quả Balance Enquiry cho toàn bộ Account của Customer được hiển thị dưới dạng JSON
API_TC_08	Nhập Account No không tồn tại trong hệ thống khi sử dụng Balance Enquiry API	1. Nhập Account No không tồn tại vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkbalanceenquiry.php?CUSTOMER_ID=68195&PASSWORD=1234!&Account_No=154	Thông báo lỗi "NoData" hiển thị dưới dạng JSON
API_TC_09	Nhập Account No hợp lệ khi sử dụng Balance Enquiry API	1. Nhập Account No hợp lệ vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkbalanceenquiry.php?CUSTOMER_ID=68195&PASSWORD=1234!&Account_No=1	Thông báo thành công "Success" và kết quả Balance Enquiry cho Account được hiển thị dưới dạng JSON



API_TC_10	Nhập Customer ID không tồn tại trong hệ thống khi sử dụng Balance Enquiry API	1. Nhập Customer ID không tồn tại vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkbalanceenquiry.php?CUSTOMER_ID=685&PASSWORD=1234!&Account_No=154	Thông báo lỗi "Login Credentials Incorrect" hiển thị dưới dạng JSON
API_TC_11	Không nhập Password khi sử dụng Balance Enquiry API	1. Không nhập Password vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkbalanceenquiry.php?CUSTOMER_ID=68195&PASSWORD=&Account_No=154	Thông báo lỗi "Login Credentials Incorrect" hiển thị dưới dạng JSON
API_TC_12	Nhập sai Password khi sử dụng Balance Enquiry API	1. Nhập Password sai vào URL 2. Nhấn Enter để thực hiện	demo.guru99.com/V4/sinkbalanceenquiry.php?CUSTOMER_ID=68195&PASSWORD=12&&Account_No=154	Thông báo lỗi "Login Credentials Incorrect" hiển thị dưới dạng JSON

*Bảng 2.16. Danh sách test cases API*

## 2.3. Kết quả kiểm thử

Bản báo cáo kiểm thử đầy đủ và danh sách bug được viết chi tiết lần lượt tại sheet *Báo cáo kiểm thử* và *Bug* trong tài liệu thiết kế test case tại PHỤ LỤC.

### 2.3.1. Kết quả thực hiện kiểm thử thủ công

STT	Tên chức năng	Pass	Fail	Untested	N/A	Tổng số test case
1	New Account	7	1	0	0	8
2	New Customer	35	0	0	0	35
3	Balance Enquiry	4	0	0	0	4
4	Customized Statement	10	2	0	0	12
5	Delete Account	4	0	0	0	4
6	Delete Customer	5	0	0	0	5
7	Deposit	10	0	0	0	10
8	Edit Account	4	0	0	0	4
9	Edit Customer	35	0	0	0	35
10	Fund Transfer	13	1	0	0	14
11	Login	4	0	0	0	4
12	Mini Statement	4	0	0	0	4
13	Change Password	9	0	0	0	9
14	Withdraw	10	0	0	0	10
15	API	10	2	0	0	12
	<b>Tổng số</b>	<b>164</b>	<b>6</b>	<b>0</b>	<b>0</b>	<b>170</b>

*Bảng 2.17. Kết quả kiểm thử thủ công*



### 2.3.2. Quản lý bug

#### a) Thông kê bug

STT	Tên chức năng	Critical	Major	Minor	Tổng số bug
1	New Account	0	1	0	1
2	Customized Statement	0	2	0	2
3	Withdraw	0	1	0	1
4	API	0	2	0	2
<b>Tổng số</b>		<b>0</b>	<b>6</b>	<b>0</b>	<b>6</b>

Bảng 2.18. Bảng thống kê Bug dựa trên mức độ nghiêm trọng và chức năng

#### b) Chi tiết bug

Bug ID	Mô tả chi tiết bug	Các bước thực hiện	Kết quả mong đợi	Kết quả thực tế	Test case ID	Mức độ nghiêm trọng lỗi
BUG_001	Thông báo lỗi không hiển thị	1. Nhập 499 vào trường [Initial Deposit] 2. Nhấn Tab để đến trường tiếp theo	Hiển thị thông báo lỗi	Thông báo lỗi không hiển thị	NA_TC_07	Major
BUG_002	Thông báo lỗi không hiển thị	1. Không nhập dữ liệu vào trường [Minimum Transaction Value] 2. Nhấn Tab để đến trường tiếp theo	Thông báo lỗi "Amount Lower Limit is required" hiển thị	Thông báo lỗi không hiển thị	CS_TC_05	Major
BUG_003	Thông báo lỗi không hiển thị	1. Không nhập dữ liệu vào trường [Number of Transaction] 2. Nhấn Tab để đến trường tiếp theo	Thông báo lỗi "Minimum Transaction Value must not be blank" hiển thị	Thông báo lỗi không hiển thị	CS_TC_09	Major
BUG_004	Không thể thực hiện API	1. Không nhập Account No vào URL thực hiện Balance Enquiry API 2. Nhấn Enter	Thông báo thành công "Success" và kết quả Balance Enquiry cho toàn bộ Account của Customer được hiển thị dưới dạng JSON	Không thể thực hiện API	API_TC_07	Major
BUG_005	Không thể thực hiện API	1. Nhập Account No hợp lệ vào URL thực hiện Balance Enquiry API 2. Nhấn Enter	Thông báo thành công "Success" và kết quả Balance Enquiry cho Account được hiển thị dưới dạng JSON	Không thể thực hiện API	API_TC_09	Major
BUG_006	Thông báo lỗi hiển thị không như mong đợi	1. Nhập Account No hợp lệ vào URL thực hiện Balance Enquiry API 2. Nhấn Enter	Thông báo lỗi "Account No must not be blank" hiển thị	Thông báo lỗi hiển thị "Account Number must not be blank"	WD_TC_01	Major

Bảng 2.19. Danh sách Bug

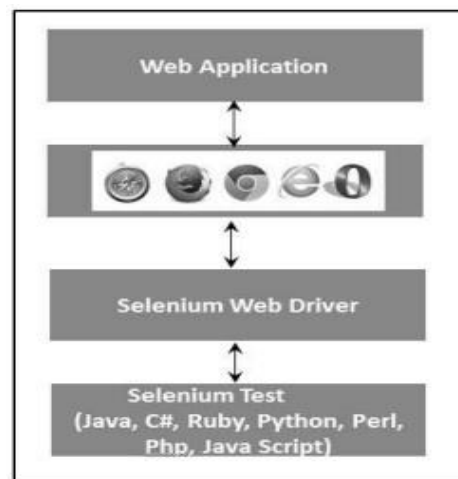
## CHƯƠNG 3. THỰC HIỆN KIỂM THỬ TỰ ĐỘNG

### 3.1. Công cụ sử dụng

#### 3.1.1. *Selenium WebDriver*

##### a) Cách thức hoạt động

Selenium WebDriver giúp người dùng tương tác với các ứng dụng web thông qua các phiên bản Driver tương ứng với nhiều trình duyệt khác nhau bằng các câu lệnh được cung cấp trong thư viện Selenium. Cụ thể, Selenium WebDriver hiện hỗ trợ cho các trình duyệt: *Chrome, Edge, FireFox, Safari*.



Hình 3.1. Mô hình hoạt động của Selenium Web Driver

##### b) Cú pháp câu lệnh

Một số câu lệnh thường dùng:

- *element.sendKeys()* – sử dụng để nhập văn bản hoặc thực hiện mô phỏng nhấn các nút trên bàn phím vào element đang được trỏ đến
- *element.clear()* – sử dụng để xóa nội dung của input box đang được trỏ đến thông qua biến element
- *driver.switchTo()* – sử dụng để di chuyển giữa các cửa sổ trong trình duyệt, được kết hợp với các hàm *window("windowName")*, *frame("frameName")*, *alert()*, ...
- *driver.navigate()* – sử dụng để điều hướng driver, có thể kết hợp với *to("URL")* để chuyển hướng đến URL, hoặc dùng chung với *forward()*, *back()* để chuyển hướng đến trang tiếp theo hoặc trang trước đó
- *driver.close()* – sử dụng để đóng trình duyệt hiện tại
- *driver.quit()* – sử dụng để đóng tất cả cửa sổ
- *driver.refresh()* – sử dụng để tải lại trang hiện tại
- *driver.findElement(...)* – sử dụng để tìm kiếm web element trong trang web, nhận input là các locators, bao gồm: class name, css selector, id, name, link text, partial link text, tag name, xpath

### 3.1.2. *TestNG*

TestNG là framework được sử dụng trong kiểm thử nhằm đơn giản hóa các nhu cầu kiểm thử, từ unit testing đến integration testing.

Để sử dụng TestNG trong kiểm thử, người dùng sau khi viết test script cần thêm các TestNG annotations, có thể thêm groups, ... để cung cấp thông tin trong tệp .xml.

Một số annotations thường dùng:

- *@Test* – annotation bắt buộc sử dụng để đánh dấu các test script
- *@BeforeXXX* – annotation được sử dụng để đánh dấu các đoạn code cần thực hiện trước khi thực hiện một test case hoặc một test class với XXX là Method hay Class
- *@AfterXXX* – tương tự với *@BeforeXXX* nhưng annotation này được sử dụng để đánh dấu các đoạn code cần thực hiện sau khi thực hiện test

### 3.1.3. *RestAssured*

RestAssured là một thư viện của ngôn ngữ lập trình Java, được sử dụng để thực hiện kiểm thử tự động RESTful API bằng việc kiểm tra các mong đợi đối với các HTTP Response, hỗ trợ các yêu cầu POST, GET, PUT, DELETE, ... và cho phép người dùng truyền giá trị các biến (*parameter*) vào các request.

Một số hàm thường dùng của RestAssured:

- .params()* – Sử dụng để truyền giá trị cho các biến đầu vào của API
- .get()* / *.post()* – Sử dụng để thực hiện request GET/POST
- .getStatusCode()* – Sử dụng để lấy mã trạng thái của HTTP response

## 3.2. Thiết kế Test Script

### 3.2.1. *Phạm vi kiểm thử*

Kiểm thử tự động được thực hiện đối với các chức năng:

- API
- Login
- Withdraw
- Change Password
- Fund Transfer

### 3.2.2. Page Objects

Các Test Script phục vụ cho việc kiểm thử tự động được áp dụng mô hình **Page Object Model (POM)**. POM được áp dụng thông qua tạo các *Class* tương ứng cho từng trang web, với mỗi *Class* chứa các *locator*, *element*, *method* được định nghĩa sẵn để các test script có thể sử dụng nhằm thực hiện các hành động cụ thể.

- a) GeneralPage – class chứa các element, method dùng chung của các trang

#### *Locators*

```
private final By tabWithdraw = By.xpath("//a[text()='Withdrawal']");
private final By tabChangePassword = By.xpath("//a[text()='Change Password']");
private final By tabFundTransfer = By.xpath("//a[text()='Fund Transfer']");
```

#### *Elements*

```
protected WebElement getTabWithdraw(WebDriver driver) {
    return driver.findElement(tabWithdraw);
}
protected WebElement getTabChangePassword(WebDriver driver) {
    return driver.findElement(tabChangePassword);
}
protected WebElement getTabFundTransfer(WebDriver driver) {
    return driver.findElement(tabFundTransfer);
}
```

#### *Methods*

```
public WithdrawPage gotoWithdraw(WebDriver driver) {
    WebElement element = this.getTabWithdraw(driver);
    new Actions(driver)
        .moveToElement(element)
        .click()
        .perform();
    this.adsRemove(driver);
    return new WithdrawPage();
}
public ChangePasswordPage gotoChangePassword(WebDriver driver) {
    WebElement element = this.getTabChangePassword(driver);
    new Actions(driver)
        .moveToElement(element)
        .click()
        .perform();
    this.adsRemove(driver);
    return new ChangePasswordPage();
}
public FundTransferPage gotoFundTransfer(WebDriver driver) {
    WebElement element = this.getTabFundTransfer(driver);
    new Actions(driver)
        .moveToElement(element)
        .click()
        .perform();
    this.adsRemove(driver);
    return new FundTransferPage();
}
```

```

protected void adsRemove(WebDriver driver) {
    if (driver.getCurrentUrl().contains("#google_vignette")) {
        try {
            Thread.sleep(3000);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        WebElement frame1 =
driver.findElement(By.id("google_ads_iframe_/24132379/INTERSTITIAL_DemoGuru99_0"));
        driver.switchTo().frame(frame1);
        try {
            driver.findElement(By.xpath("//div[@id='dismiss-
button']")).click();
        } catch (NoSuchElementException e) {
            WebElement frame2 = driver.findElement(By.id("ad_iframe"));
            driver.switchTo().frame(frame2);
            driver.findElement(By.xpath("//div[@id='dismiss-
button']")).click();
        }
        driver.switchTo().defaultContent();
    }
}

```

## b) LoginPage

### *Locators*

```

private final By fieldUserID = By.name("uid");
private final By labelUserID = By.id("message23");
private final By fieldPassword = By.name("password");
private final By labelPassword = By.id("message18");

```

### *Elements*

```

private WebElement getFieldUserID(WebDriver driver) {
    return driver.findElement(fieldUserID);
}
private WebElement getFieldPassword(WebDriver driver) {
    return driver.findElement(fieldPassword);
}
private WebElement getLabelUserID(WebDriver driver) {
    return driver.findElement(labelUserID);
}
private WebElement getLabelPassword(WebDriver driver) {
    return driver.findElement(labelPassword);
}

```

### *Methods*

```

public GeneralPage successLogin(WebDriver driver, String userID, String password) {
    WebElement textField = this.getFieldUserID(driver);

    new Actions(driver)
        .sendKeys(textField, userID)
        .sendKeys(Keys.TAB)
        .sendKeys(password)
        .sendKeys(Keys.TAB)
        .sendKeys(Keys.ENTER)
        .perform();
    return new GeneralPage();
}

```

```

public String getUserIDMessage(WebDriver driver, String userID) {
    WebElement textField = this.getFieldUserID(driver);

    new Actions(driver)
        .sendKeys(textField, userID)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelUserID(driver).getText();
}
public String getPasswordMessage(WebDriver driver, String password) {
    WebElement textField = this.getFieldPassword(driver);
    new Actions(driver)
        .sendKeys(textField, password)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelPassword(driver).getText();
}

```

### c) WithdrawPage

#### *Locators*

```

private final By fieldAccountNo = By.name("accountno");
private final By labelAccountNo = By.id("message2");
private final By fieldAmount = By.name("ammount");
private final By labelAmount = By.id("message1");
private final By fieldDescription = By.name("desc");
private final By labelDescription = By.id("message17");

```

#### *Elements*

```

private WebElement getFieldAccountNo(WebDriver driver) {
    return driver.findElement(fieldAccountNo);
}
private WebElement getFieldAmount(WebDriver driver) {
    return driver.findElement(fieldAmount);
}
private WebElement getFieldDescription(WebDriver driver) {
    return driver.findElement(fieldDescription);
}
private WebElement getLabelAccountNo(WebDriver driver) {
    return driver.findElement(labelAccountNo);
}
private WebElement getLabelAmount(WebDriver driver) {
    return driver.findElement(labelAmount);
}
private WebElement getLabelDescription(WebDriver driver) {
    return driver.findElement(labelDescription);
}

```

#### *Methods*

```

public String getAccountNoMessage(WebDriver driver, String accountNo) {
    WebElement textField = this.getFieldAccountNo(driver);
    new Actions(driver)
        .sendKeys(textField, accountNo)
        .sendKeys(Keys.TAB)
        .perform();
    return this.getLabelAccountNo(driver).getText();
}

```

```

public String getAmountMessage(WebDriver driver, String amount) {
    WebElement textField = this.getFieldAmount(driver);

    new Actions(driver)
        .sendKeys(textField, amount)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelAmount(driver).getText();
}
public String getDescriptionMessage(WebDriver driver, String description) {
    WebElement textField = this.getFieldDescription(driver);
    new Actions(driver)
        .sendKeys(textField, description)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelDescription(driver).getText();
}

```

#### d) ChangePasswordPage

##### *Locators*

```

private final By fieldOldPassword = By.name("oldpassword");
private final By labelOldPassword = By.id("message20");
private final By fieldNewPassword = By.name("newpassword");
private final By labelNewPassword = By.id("message21");
private final By fieldConfirmPassword = By.name("confirmpassword");
private final By labelConfirmPassword = By.id("message22");

```

##### *Elements*

```

private WebElement getFieldOldPassword(WebDriver driver) {
    return driver.findElement(fieldOldPassword);
}
private WebElement getFieldNewPassword(WebDriver driver) {
    return driver.findElement(fieldNewPassword);
}
private WebElement getFieldConfirmPassword(WebDriver driver) {
    return driver.findElement(fieldConfirmPassword);
}
private WebElement getLabelOldPassword(WebDriver driver) {
    return driver.findElement(labelOldPassword);
}
private WebElement getLabelNewPassword(WebDriver driver) {
    return driver.findElement(labelNewPassword);
}
private WebElement getLabelConfirmPassword(WebDriver driver) {
    return driver.findElement(labelConfirmPassword);
}

```

##### *Methods*

```

public String getOldPasswordMessage(WebDriver driver, String oldPassword) {
    WebElement textField = this.getFieldOldPassword(driver);
    new Actions(driver)
        .sendKeys(textField, oldPassword)
        .sendKeys(Keys.TAB)
        .perform();
    return this.getLabelOldPassword(driver).getText();
}

```

```

public String getNewPasswordMessage(WebDriver driver, String newPassword) {
    WebElement textField = this.getFieldNewPassword(driver);
    new Actions(driver)
        .sendKeys(textField, newPassword)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelNewPassword(driver).getText();
}

public String getConfirmPasswordMessage(WebDriver driver, String newPassword,
String confirmPassword) {
    WebElement textField = this.getFieldNewPassword(driver);
    new Actions(driver)
        .sendKeys(textField, newPassword)
        .sendKeys(Keys.TAB)
        .sendKeys(confirmPassword)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelConfirmPassword(driver).getText();
}

```

#### e) FundTransferPage

##### *Locators*

```

private final By fieldPayersNo = By.name("payersaccount");
private final By labelPayersNo = By.id("message10");
private final By fieldPayeesNo = By.name("payeeaccount");
private final By labelPayeesNo = By.id("message11");
private final By fieldAmount = By.name("amount");
private final By labelAmount = By.id("message1");
private final By fieldDescription = By.name("desc");
private final By labelDescription = By.id("message17");

```

##### *Elements*

```

private WebElement getFieldPayersNo(WebDriver driver) {
    return driver.findElement(fieldPayersNo);
}

private WebElement getFieldPayeesNo(WebDriver driver) {
    return driver.findElement(fieldPayeesNo);
}

private WebElement getFieldAmount(WebDriver driver) {
    return driver.findElement(fieldAmount);
}

private WebElement getFieldDescription(WebDriver driver) {
    return driver.findElement(fieldDescription);
}

private WebElement getLabelPayersNo(WebDriver driver) {
    return driver.findElement(labelPayersNo);
}

private WebElement getLabelPayeesNo(WebDriver driver) {
    return driver.findElement(labelPayeesNo);
}

private WebElement getLabelAmount(WebDriver driver) {
    return driver.findElement(labelAmount);
}

private WebElement getLabelDescription(WebDriver driver) {
    return driver.findElement(labelDescription);
}

```



## Methods

```
public String getPayersNoMessage(WebDriver driver, String payersNo) {
    WebElement textField = this.getFieldPayersNo(driver);
    new Actions(driver)
        .sendKeys(textField, payersNo)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelPayersNo(driver).getText();
}

public String getPayeesNoMessage(WebDriver driver, String payeesNo) {
    WebElement textField = this.getFieldPayeesNo(driver);
    new Actions(driver)
        .sendKeys(textField, payeesNo)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelPayeesNo(driver).getText();
}

public String getAmountMessage(WebDriver driver, String amount) {
    WebElement textField = this.getFieldAmount(driver);
    new Actions(driver)
        .sendKeys(textField, amount)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelAmount(driver).getText();
}

public String getDescriptionMessage(WebDriver driver, String description) {
    WebElement textField = this.getFieldDescription(driver);
    new Actions(driver)
        .sendKeys(textField, description)
        .sendKeys(Keys.TAB)
        .perform();

    return this.getLabelDescription(driver).getText();
}
```

### 3.2.3. Test Scripts

#### a) API

#### *Phương thức dùng chung các test case*

```
private Response getResponse(String URL, String customerID, String password,
String accountNo) {
    RestAssured.baseURI = baseUrl;
    RequestSpecification request = RestAssured.given()
        .params("CUSTOMER_ID", customerID,
            "PASSWORD", password,
            "Account_No", accountNo);
    return request.get(URL);
}

private String formatJSON(String jsonString) {
    return jsonString.replaceAll("&quot;", "\"").replaceAll(":\\":", \"\\":");
}

private String getErrorMessage(Response response) {
    String json = formatJSON(response.asString());
    JsonPath jsonPath = new JsonPath(json).setRootPath("message");
    return jsonPath.get("ErrorMsg");
}
```

Việc kiểm thử tự động API được thực hiện theo các bước:

1. Gửi request, nhận response – Sử dụng hàm `getResponse()` định nghĩa above
2. Kiểm tra status code để đảm bảo API được thực hiện
3. Kiểm tra ErrorMessage đúng với tài liệu đặc tả – Sử dụng hàm `getErrorMessage()` định nghĩa above

### *Một số test case*

Các test script trước khi bắt đầu kiểm thử sẽ thực hiện thông báo *Test case ID* và *mô tả chi tiết* để theo dõi trong quá trình kiểm thử, được thiết kế dựa trên *Bảng 2.16. Danh sách test cases API*.

```
@Test
public void API_TC_01() {
    System.out.println("API_TC_01 - Không nhập Account No khi sử dụng Mini
Statement API");
    Response response = getResponse("/sinkministatement.php",
validCustomerID, validPassword, "");
    int actualStatusCode = response.getStatusCode();
    Assert.assertEquals(actualStatusCode, expectedStatusCode, "Không thể thực
hiện API");
    String actualErrorMsg = getErrorMessage(response);
    String expectedErrorMsg = "NoData";
    Assert.assertEquals(actualErrorMsg, expectedErrorMsg, "Thông báo lỗi hiển
thị không như mong đợi");
}

@Test
public void API_TC_09() {
    System.out.println("API_TC_09 - Nhập Account No hợp lệ khi sử dụng
Balance Enquiry API");
    Response response = getResponse("/sinkbalanceenquiry.php",
validCustomerID, validPassword, validAccountNo);
    int actualStatusCode = response.getStatusCode();
    Assert.assertEquals(actualStatusCode, expectedStatusCode, "Không thể thực
hiện API");
    String actualErrorMsg = getErrorMessage(response);
    String expectedErrorMsg = "Success";
    Assert.assertEquals(actualErrorMsg, expectedErrorMsg, "Thông báo lỗi hiển
thị không như mong đợi");
}

@Test
public void API_TC_10() {
    System.out.println("API_TC_10 - Nhập Customer ID không tồn tại trong hệ
thống khi sử dụng Balance Enquiry API");
    Response response = getResponse("/sinkbalanceenquiry.php",
invalidCustomerID, validPassword, validAccountNo);
    int actualStatusCode = response.getStatusCode();
    Assert.assertEquals(actualStatusCode, expectedStatusCode, "Không thể thực
hiện API");
    String actualErrorMsg = getErrorMessage(response);
    String expectedErrorMsg = "Login Credentials Incorrect";
    Assert.assertEquals(actualErrorMsg, expectedErrorMsg, "Thông báo lỗi hiển
thị không như mong đợi");
}
```

## b) Chức năng Login

### *Phương thức Before và After*

Phương thức Before được sử dụng để khởi động trình duyệt trước khi bắt đầu mỗi test case, thực hiện truy cập trang web <https://demo.guru99.com/V4/> được lưu trong biến baseURL.

```
@BeforeMethod
public void beforeMethod() {
    edgeDriver = new EdgeDriver();
    edgeDriver.manage().window().maximize();
    edgeDriver.navigate().to(baseURL);
    loginPage = new LoginPage();
}
```

Phương thức After được sử dụng để tắt trình duyệt sau mỗi test case khi đã thực hiện kiểm thử xong.

```
@AfterMethod
public void afterMethod() {
    edgeDriver.quit();
}
```

### *Các test case*

Các test script trước khi bắt đầu kiểm thử sẽ thực hiện thông báo *Test case ID* và mô tả chi tiết để theo dõi trong quá trình kiểm thử, được thiết kế dựa trên Bảng 2.12. Danh sách test cases chức năng Login.

```
@Test
public void LI_TC_01() {
    System.out.println("LI_TC_01 - Không nhập User ID");
    String expectedMessage = "User-ID must not be blank";
    String actualMessage = loginPage.getUserIDMessage(edgeDriver, "");
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo hiển thị không như mong đợi");
}

@Test
public void LI_TC_02() {
    System.out.println("LI_TC_02 - Nhập User ID hợp lệ");
    String expectedMessage = "";
    String actualMessage = loginPage.getUserIDMessage(edgeDriver, validUserID);
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo hiển thị không như mong đợi");
}

@Test
public void LI_TC_03() {
    System.out.println("LI_TC_03 - Không nhập Password");
    String expectedMessage = "Password must not be blank";
    String actualMessage = loginPage.getPasswordMessage(edgeDriver, "");
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo lỗi hiển thị không như mong đợi");
}
```

### c) Chức năng Withdraw

#### *Phương thức Before và After*

Phương thức Before được sử dụng để khởi động trình duyệt trước khi bắt đầu mỗi test case, thực hiện truy cập trang web <https://demo.guru99.com/V4/> được lưu trong biến baseURL, thực hiện đăng nhập vào trang web và điều hướng đến giao diện Withdraw.

```
@BeforeMethod
public void beforeMethod() {
    edgeDriver = new EdgeDriver();
    edgeDriver.manage().window().maximize();
    edgeDriver.navigate().to(baseURL);
    LoginPage loginPage = new LoginPage();
    GeneralPage generalPage = loginPage.successLogin(edgeDriver, validUserID,
    validPassword);
    withdrawPage = generalPage.gotoWithdraw(edgeDriver);
}
```

Phương thức After được sử dụng để tắt trình duyệt sau mỗi test case khi đã thực hiện kiểm thử xong.

```
@AfterMethod
public void afterMethod() { edgeDriver.quit(); }
```

#### *Một số test case*

Các test script trước khi bắt đầu kiểm thử sẽ thực hiện thông báo *Test case ID* và mô tả chi tiết để theo dõi trong quá trình kiểm thử, được thiết kế dựa trên Bảng 2.15. Danh sách test cases chức năng Withdraw.

```
@Test
public void WD_TC_02() {
    System.out.println("WD_TC_02 - Nhập Account No chứa ký tự đặc biệt");
    String expectedMessage = "Special characters are not allowed";
    String actualMessage = withdrawPage.getAccountNoMessage(edgeDriver,
    "1245!@#");
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo lỗi hiển
    thị không như mong đợi");
}

@Test
public void WD_TC_05() {
    System.out.println("WD_TC_05 - Không nhập Amount");
    String expectedMessage = "Amount field must not be blank";
    String actualMessage = withdrawPage.getAmountMessage(edgeDriver, "");
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo lỗi hiển
    thị không như mong đợi");
}

@Test
public void WD_TC_10() {
    System.out.println("WD_TC_10 - Nhập Description hợp lệ");
    String expectedMessage = "";
    String actualMessage = withdrawPage.getDescriptionMessage(edgeDriver,
    "withdrew");
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo lỗi hiển
    thị không như mong đợi");
}
```

#### d) Chức năng Change Password

##### *Phương thức Before và After*

Phương thức Before được sử dụng để khởi động trình duyệt trước khi bắt đầu mỗi test case, thực hiện truy cập trang web <https://demo.guru99.com/V4/> được lưu trong biến baseURL, thực hiện đăng nhập vào trang web và điều hướng đến giao diện Change Password.

```
@BeforeMethod
public void beforeMethod() {
    edgeDriver = new EdgeDriver();
    edgeDriver.manage().window().maximize();
    edgeDriver.navigate().to(baseURL);
    LoginPage loginPage = new LoginPage();
    GeneralPage generalPage = loginPage.successLogin(edgeDriver, validUserID,
    validPassword);
    changePasswordPage = generalPage.gotoChangePassword(edgeDriver);
}
```

Phương thức After được sử dụng để tắt trình duyệt sau mỗi test case khi đã thực hiện kiểm thử xong.

```
@AfterMethod
public void afterMethod() { edgeDriver.quit(); }
```

##### *Một số test case*

Các test script trước khi bắt đầu kiểm thử sẽ thực hiện thông báo *Test case ID* và mô tả chi tiết để theo dõi trong quá trình kiểm thử, được thiết kế dựa trên Bảng 2.14. Danh sách test cases chức năng Change Password.

```
@Test
public void CP_TC_04() {
    System.out.println("CP_TC_04 - Nhập New Password không chứa ký tự số");
    String expectedMessage = "Enter at-least one numeric value";
    String newPassword = "abcd!@#$";
    String actualMessage =
    changePasswordPage.getNewPasswordMessage(edgeDriver, newPassword);
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo lỗi hiển
    thị không như mong đợi");
}

@Test
public void CP_TC_08() {
    System.out.println("CP_TC_08 - Nhập Confirm Password không trùng với New
    Password");
    String expectedMessage = "Passwords do not Match";
    String newPassword = "1234!@#$";
    String confirmPassword = "1234";
    String actualMessage =
    changePasswordPage.getConfirmPasswordMessage(edgeDriver, newPassword,
    confirmPassword);
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo lỗi hiển
    thị không như mong đợi");
}
```

## e) Chức năng Fund Transfer

### *Phương thức Before và After*

Phương thức Before được sử dụng để khởi động trình duyệt trước khi bắt đầu mỗi test case, thực hiện truy cập trang web <https://demo.guru99.com/V4/> được lưu trong biến baseUrl, thực hiện đăng nhập vào trang web và điều hướng đến giao diện Fund Transfer.

```
@BeforeMethod
public void beforeMethod() {
    edgeDriver = new EdgeDriver();
    edgeDriver.manage().window().maximize();
    edgeDriver.navigate().to(baseUrl);
    LoginPage loginPage = new LoginPage();
    GeneralPage generalPage = loginPage.successLogin(edgeDriver, validUserID,
    validPassword);
    fundTransferPage = generalPage.gotoFundTransfer(edgeDriver);
}
```

Phương thức After được sử dụng để tắt trình duyệt sau mỗi test case khi đã thực hiện kiểm thử xong.

```
@AfterMethod
public void afterMethod() { edgeDriver.quit(); }
```

### *Một số test case*

Các test script trước khi bắt đầu kiểm thử sẽ thực hiện thông báo *Test case ID* và mô tả chi tiết để theo dõi trong quá trình kiểm thử, được thiết kế dựa trên Bảng 2.11. Danh sách test cases chức năng Fund Transfer.

```
@Test
public void FT_TC_02() {
    System.out.println("FT_TC_02 - Nhập Payers Account Number chứa ký tự đặc biệt");
    String expectedMessage = "Special characters are not allowed";
    String payersNo = "1245!@#";
    String actualMessage = fundTransferPage.getPayersNoMessage(edgeDriver, payersNo);
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo lỗi hiển thị không như mong đợi");
}

@Test
public void FT_TC_05() {
    System.out.println("FT_TC_05 - Không nhập Payees Account Number");
    String expectedMessage = "Payees Account Number must not be blank";
    String actualMessage = fundTransferPage.getPayeesNoMessage(edgeDriver, "");
    Assert.assertEquals(actualMessage, expectedMessage, "Thông báo lỗi hiển thị không như mong đợi");
}
```

Ngoài các test script above, các test script, page object đầy đủ được đặt lần lượt tại /src/test/java/ và /src/main/java/ trong project được dẫn link tại PHỤ LỤC.

### 3.3. Kết quả kiểm thử

Bản báo cáo đầy đủ được dẫn link tại PHỤ LỤC.

#### 3.3.1. API

Class	Method	Start	Time (ms)
allTestcase			
software_interface — failed			
API.SoftwareInterfaceTest	API_TC_07	1722299738418	942
	API_TC_09	1722299740426	929
software_interface — passed			
API.SoftwareInterfaceTest	API_TC_01	1722299716156	16954
	API_TC_02	1722299733113	1105
	API_TC_03	1722299734220	1061
	API_TC_04	1722299735281	1036
	API_TC_05	1722299736321	1057
	API_TC_06	1722299737379	1036
	API_TC_08	1722299739362	1058
	API_TC_10	1722299741359	1002
	API_TC_11	1722299742362	1000
	API_TC_12	1722299743363	1014

Hình 3.2. Kết quả kiểm thử tự động API

#### Chi tiết test case fail

##### API.SoftwareInterfaceTest#API\_TC\_07

Exception
java.lang.AssertionError: Không thể thực hiện API expected [200] but found [500] at API.SoftwareInterfaceTest.API_TC_07(SoftwareInterfaceTest.java:109) at java.base/java.util.ArrayList.forEach(ArrayList.java:1541) ... Removed 37 stack frames

Hình 3.3. Chi tiết lỗi test case API\_TC\_07

##### API.SoftwareInterfaceTest#API\_TC\_09

Exception
java.lang.AssertionError: Không thể thực hiện API expected [200] but found [500] at API.SoftwareInterfaceTest.API_TC_09(SoftwareInterfaceTest.java:129) at java.base/java.util.ArrayList.forEach(ArrayList.java:1541) ... Removed 37 stack frames

Hình 3.4. Chi tiết lỗi test case API\_TC\_09

### 3.3.2. Ứng dụng web

Class	Method	Start	Time (ms)
allTestcase			
web_application — failed			
WebApplication.WithdrawPageTest	WD_TC_01	1722295848146	1631
web_application — passed			
WebApplication.ChangePasswordPageTest	CP_TC_01	1722297383423	914
	CP_TC_02	1722297429485	3306
	CP_TC_03	1722297509570	1937
	CP_TC_04	1722297585698	1931
	CP_TC_05	1722297712371	1651
	CP_TC_06	1722297839290	2504
	CP_TC_07	1722297985684	707
	CP_TC_08	1722298097261	2300
	CP_TC_09	1722298202390	2429
WebApplication.FundTransferPageTest	FT_TC_01	1722298292108	2188
	FT_TC_02	1722298428634	1491
	FT_TC_03	1722298549707	1981
	FT_TC_04	1722298644822	1070
	FT_TC_05	1722298759271	2633
	FT_TC_06	1722298847672	2014
	FT_TC_07	1722298982874	2322
	FT_TC_08	1722299105436	2115
	FT_TC_09	1722299205965	667
	FT_TC_10	1722299273095	2709
	FT_TC_11	1722299368967	1834
	FT_TC_12	1722299477830	4406
	FT_TC_13	1722299610478	1385
	FT_TC_14	1722299699730	2175
WebApplication.LoginPageTest	LI_TC_01	1722295662683	5647
	LI_TC_02	1722295734903	3580
	LI_TC_03	1722295761276	1330
	LI_TC_04	1722295781437	4830
WebApplication.WithdrawPageTest	WD_TC_02	1722295976018	1022
	WD_TC_03	1722296096476	3384
	WD_TC_04	1722296222178	1589
	WD_TC_05	1722296304336	27693
	WD_TC_06	1722296697517	19190
	WD_TC_07	1722296882954	6090
	WD_TC_08	1722297018102	1949
	WD_TC_09	1722297174754	3490
	WD_TC_10	172229727276	1504

Hình 3.5. Kết quả kiểm thử tự động các chức năng Login, Withdraw, Change Password, Fund Transfer

#### Chi tiết test case fail

WebApplication.WithdrawPageTest#WD\_TC\_01

Exception
<pre>java.lang.AssertionError: Thông báo lỗi hiển thị không như mong đợi. expected [Account No must not be blank] but found [Account Number must not be blank]     at WebApplication.WithdrawPageTest.WD_TC_01(WebApplication.java:46)     at java.base/java.util.ArrayList.forEach(ArrayList.java:1541)     ... Removed 36 stack frames</pre>

Hình 3.6. Chi tiết lỗi test case WD\_TC\_01



## CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

- **Kết quả đạt được:**

- Kiến thức đã học được:
  - Hiểu rõ về các bước trong quy trình kiểm thử
  - Hiểu được các loại kiểm thử và ứng dụng của chúng
- Kỹ năng đã đạt được:
  - Kỹ năng đọc hiểu và phân tích tài liệu đặc tả
  - Kỹ năng áp dụng các phương pháp, kỹ thuật trong thiết kế test case
  - Kỹ năng thực hiện kiểm thử và báo cáo bug
  - Kỹ năng thiết kế test case và test script

- **Hạn chế:**

- Chưa thể thực hiện kiểm thử được toàn bộ hệ thống
- Chưa thể thực hiện một số kỹ thuật test về bảo mật, hiệu suất
- Các test case, test script còn nhiều thiếu sót trong quá trình kiểm thử

- **Hướng phát triển:**

- Tìm kiếm các cơ hội khác để được học hỏi, trải nghiệm kiểm thử thêm các test case phức tạp hơn
- Học thêm về các kỹ thuật kiểm thử bảo mật, hiệu suất
- Cải thiện các điểm thiếu sót đã gặp trong quá trình thực hiện đề tài

## CHƯƠNG 5. CHECK LIST CỦA BÁO CÁO

STT	Nội dung công việc	Có	Không	Ghi chú
1	Báo cáo được trình bày (định dạng) đúng với yêu cầu.	X		
2	Báo cáo có số lượng trang đáp ứng đúng yêu cầu (30-50 trang)	X		
3	Báo cáo trình bày được phân mở đầu bao gồm: Mục tiêu, Phạm vi và đối tượng, kết cấu ...	X		
4	Báo cáo trình bày về công ty, vị trí việc làm (công việc đó làm gì, kiến thức và kỹ năng cần thiết là gì, con đường phát triển sự nghiệp (career path)), cơ sở lý thuyết phù hợp với nội dung của đề tài (Tối đa 10-12 trang)	X		
5	Báo cáo có sản phẩm cụ thể phù hợp với mục tiêu đặt ra của đề tài	X		
6	Báo cáo có phần kết luận và hướng phát triển của đề tài	X		

## CHƯƠNG 6. PHỤ LỤC

### *Tài liệu đặc tả của hệ thống*

<https://docs.google.com/document/d/1PZQZKt7hqS417QjYRMppPnTwfj8V54XUA7nZUnYvumE/edit?usp=sharing>

### *Tài liệu thiết kế Test Case*

<https://docs.google.com/spreadsheets/d/1Xe4g3rAjkJN-ssRFNZGA65m8F5EN3V7ZB-MP9lVemrc/edit?usp=sharing>

### *Project files*

<https://github.com/ThyHuong/Guru99Bank>

### *Báo cáo thực hiện kiểm thử tự động đầy đủ*

[https://drive.google.com/file/d/1Ye8x0klbFxBHYV7P6zZpivKkAatjQ\\_9c/view?usp=sharing](https://drive.google.com/file/d/1Ye8x0klbFxBHYV7P6zZpivKkAatjQ_9c/view?usp=sharing)