

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**TỰ ĐỘNG SINH BỘ KIỂM THỬ DỰA TRÊN TÀI LIỆU
ĐẶC TẢ YÊU CẦU NGHIỆP VỤ SRS**

Tác giả: Bùi Thị Thúy

**LUẬN VĂN THẠC SĨ
Chuyên ngành: HỆ THỐNG THÔNG TIN**

Hà Nội, 10/2016

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**TỰ ĐỘNG SINH BỘ KIỂM THỬ DỰA TRÊN TÀI LIỆU
ĐẶC TẢ YÊU CẦU NGHIỆP VỤ SRS**

Tác giả: Bùi Thị Thúy

Giảng viên hướng dẫn:
PGS.TS. Trương Ninh Thuận

Hà Nội, 10/2016

LỜI CAM ĐOAN

Tác giả xin cam đoan kết quả đạt được trong luận văn là sản phẩm của riêng cá nhân Tác giả và được sự hướng dẫn khoa học của PGS. TS. Trương Ninh Thuận, không sao chép lại của người khác. Trong toàn bộ nội dung của luận văn, những điều trình bày của cá nhân hoặc được tổng hợp của nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tác giả xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà Nội, ngày tháng năm 2016

HỌC VIÊN

Bùi Thị Thúy

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành và sâu sắc nhất tới PGS.TS Trương Ninh Thuận, người thầy đã trực tiếp hướng dẫn tận tình và đóng góp những ý kiến quý báu cho em trong suốt quá trình thực hiện luận văn tốt nghiệp này.

Em xin gửi lời cảm ơn đến các thầy cô giáo trường Đại học Công nghệ - Đại học Công nghệ - Đại học Quốc gia Hà Nội, đã tận tâm truyền đạt những kiến thức quý báu làm nền tảng cho em trong công việc và cuộc sống. Qua đây, em cũng xin gửi lời cảm ơn đến các đồng nghiệp tại công ty TNHH FPT Software đã giúp đỡ em trong quá trình làm thực nghiệm cho luận văn.

Cuối cùng, em xin được cảm ơn cha mẹ, người thân, bạn bè và đồng nghiệp của em tại, những người đã luôn bên em, khuyến khích và động viên em trong cuộc sống và học tập.

HỌC VIÊN

Bùi Thị Thúy

MỤC LỤC

Danh mục các ký hiệu và chữ viết tắt.....	7
Danh mục bảng.....	8
Danh mục hình vẽ.....	9
MỞ ĐẦU	10
CHƯƠNG 1. GIỚI THIỆU CHUNG	11
1.1 Nội dung luận văn	11
1.2 Cấu trúc luận văn	11
CHƯƠNG 2. CÁC KHÁI NIỆM TỔNG QUAN.....	12
2.1 Giới thiệu tổng quan về SRS.....	12
2.1.1 Khái niệm SRS	12
2.1.2 Vị trí của SRS trong quá trình xây dựng phần mềm	12
2.1.3 Cấu trúc tổng quan của SRS.....	12
2.2 Giới thiệu về Use Case.....	13
2.2.1 Khái niệm Use Case	13
2.2.2 Vai trò của Use Case trong SRS	13
2.2.3 Cấu trúc tổng quan của Use Case.....	13
2.3 Giới thiệu tổng quan về Test Case	13
2.3.1 Khái niệm Test Case	13
2.3.2 Vị trí của Test Case trong quá trình xây dựng phần mềm	14
2.3.3 Cấu trúc tổng quan Test Case.....	15
CHƯƠNG 3. GIẢI PHÁP XÂY DỰNG TEST CASE DỰA TRÊN SRS	16
3.1 Dữ liệu đầu vào	16
3.1.1 Thuộc tính của Use Case.....	16
3.1.2 Luồng hoạt động (Activity Flow)	16
3.1.3 Các quy tắc nghiệp vụ (Business Rules)	16
3.2 Dữ liệu đầu ra.....	17
3.3 Phương pháp thực hiện	17
CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG	18

4.1	POI Apache	18
4.1.1	Tính năng của Apache POI	18
4.1.2	Sử dụng Apache POI trong đọc file SRS	19
4.2	JXLS.....	21
4.2.1	Giới thiệu.....	21
4.2.2	Tính năng, đặc điểm	21
4.2.3	Sử dụng JXLS để tạo file excel.....	21
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		23
TÀI LIỆU THAM KHẢO		24
PHỤ LỤC		25

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

STT	Từ viết tắt	Nghĩa đầy đủ	Ghi chú
1	SRS	Software Specification	
2	ID	Identification	
3	POI	Poor Obfuscation Implementation	
4	HSSF	Horrible SpreadSheet Format	
5	XSSF	XML SpreadSheet Format	
6	HPSF	Horrible Property Set Format	
7	HWPf	Horrible Word Processor Format	
8	HSLF	Horrible Slide Layout Format	
9	HDGF	Horrible DiaGram Format	
10	HPBF	Horrible PuBlisher Format	
11	HSMF	Horrible Stupid Mail Format	
12	DDF	Dreadful Drawing Format	
13	XML	eXtensible Markup Language	

DANH MỤC BẢNG

Table 1: Cấu trúc của một Test Case thông thường.	14
Table 2: Thuộc tính của Use Case.....	16

DANH MỤC HÌNH VẼ

Figure 1: Vị trí của Test Case trong quá trình xây dựng phần mềm	15
--	----

MỞ ĐẦU

Ngày nay, trong các quy trình sản xuất phần mềm, ngoài khâu hình thành và xây dựng sản phẩm, các công ty chuyên về sản xuất phần mềm luôn chú trọng đến quá trình đầu vào và đầu ra của sản phẩm, bởi hai quá trình này có thể tác động một cách trực tiếp đến mục tiêu và chất lượng của một sản phẩm phần mềm.

Về quá trình đầu vào của sản phẩm, một số công ty phần mềm lớn hiện nay đã xây dựng một quy trình thu thập yêu cầu phần mềm và xây dựng một bộ tài liệu chuẩn để làm đầu vào cho quá trình coding và xây dựng sản phẩm. Đầu ra của quá trình này là một bộ tài liệu về yêu cầu phần mềm, được gọi là SRS (Software Requirement Specification). Với bộ tài liệu chuẩn này, các bên liên quan đều có thể sử dụng như một bộ tài liệu chung và chuẩn nhất, được cập nhật cũng như sử dụng xuyên suốt trong toàn bộ dự án phần mềm.

Về quá trình đầu ra của sản phẩm, hầu hết các công ty đều đã xây dựng một đội ngũ kiểm thử về chất lượng của sản phẩm, và toàn bộ quy trình hoạt động của sản phẩm để đảm bảo rằng sản phẩm phần mềm này đang được xây dựng theo đúng như yêu cầu và mục tiêu đề ra ban đầu. Hiện nay, tại các công ty phần mềm lớn và nhỏ, họ đều xây dựng một đội ngũ kiểm thử, được gọi là tester, với những khóa đào tạo chuyên nghiệp để có thể tiến hành chạy những test case sau khi sản phẩm đã hoàn thành, đảm bảo rằng sau khi sản phẩm đưa vào sử dụng sẽ đúng với mục tiêu và yêu cầu ban đầu, và tránh được những lỗi về coding, mang lại cho người sử dụng một sản phẩm tương đối hoàn hảo.

Trong quá trình kiểm thử đầu ra của sản phẩm, hiện nay tất cả các test case đều được tester viết bằng tay, sau đó sử dụng các test case đó cho việc kiểm thử. Công việc này là một công việc tương đối tốn thời gian, vì mỗi sản phẩm phần mềm thường có số lượng test case lớn, có những sản phẩm phần mềm với quy mô lớn có thể lên đến hàng chục nghìn test case, điều này vô hình chung thường mang lại những áp lực vô hình cho những người làm công việc kiểm thử phần mềm.

Từ những mong muốn và nhu cầu thiết thực trên, chúng tôi mong muốn nghiên cứu và xây dựng một sản phẩm có thể tự động chuyển hóa các thông tin từ SRS thành dạng test case, để có thể hỗ trợ cho quá trình xây dựng một bộ test case chuẩn từ các yêu cầu phần mềm, phục vụ cho quá trình kiểm thử phần mềm, và giúp tiết kiệm thời gian cho tester trong việc viết test case.

CHƯƠNG 1. GIỚI THIỆU CHUNG

1.1 Nội dung luận văn

Luận văn là một chương trình phần mềm với mục tiêu có thể sinh tự động ra một bộ Test Case dựa trên dữ liệu đầu vào là một tài liệu đặc tả yêu cầu nghiệp vụ SRS. Bộ Test Case này sẽ được sử dụng là đầu vào cho quá trình kiểm thử phần mềm trong quy trình sản xuất phần mềm.

Trong khuôn khổ luận văn này, tác giả mong muốn đề cập đến những khái niệm căn bản liên quan đến SRS và Test Case, các lý thuyết chung, và đưa ra những kết quả nghiên cứu bước đầu.

Tác giả cũng mong muốn có thể giải quyết được vấn đề sinh Test Case từ các yêu cầu phần mềm, từ đó phát triển được bộ công cụ sinh Test Case tự động, đưa ra những giải pháp công nghệ để có thể giải quyết bài toán đặt ra.

1.2 Cấu trúc luận văn

Cấu trúc của luận văn được chia thành 5 phần chính:

Chương 1: Giới thiệu tổng quan về bài toán và nội dung chính của luận văn.

Chương 2: Đưa ra các khái niệm tổng quan về các đối tượng liên quan.

Chương 3: Trình bày giải pháp sinh Test Case tự động.

Chương 4: Giới thiệu về các công nghệ sử dụng.

Chương 5: Kết luận và định hướng phát triển.

Với 5 nội dung đề cập trên, tác giả mong muốn cung cấp đầy đủ thông tin để luận văn có thể trở thành một luận văn nghiên cứu với những vấn đề được giải quyết một cách triệt để và có định hướng phát triển lâu dài.

CHƯƠNG 2. CÁC KHÁI NIỆM TỔNG QUAN

2.1 Giới thiệu tổng quan về SRS

2.1.1 Khái niệm SRS

Hiện nay, các công ty về phần mềm đều có xu hướng xây dựng một bộ tài liệu yêu cầu phần mềm chuẩn cho mỗi một dự án phần mềm, để đảm bảo rằng tất cả các bên liên quan đều có những hiểu biết đúng đắn giống nhau về mục tiêu và đầu ra của sản phẩm phần mềm đó. Trên thế giới hiện nay cũng đã có những chuẩn chung cho bộ tài liệu SRS này.

SRS là từ viết tắt của Software Requirement Specification (Tài liệu đặc tả yêu cầu phần mềm). “Một tài liệu đặc tả yêu cầu phần mềm (SRS) là một mô tả của một hệ thống phần mềm được phát triển. Nó đưa ra các yêu cầu chức năng và phi chức năng, và có thể bao gồm một tập hợp các trường hợp sử dụng để mô tả tương tác người dùng mà một sản phẩm phần mềm phải cung cấp” ^[1].

SRS thiết lập cơ sở cho một thỏa thuận giữa khách hàng và các nhà thầu hoặc nhà cung cấp và các bên liên quan (trong một số dự án định hướng thị trường, các bên liên quan có thể là các đơn vị tiếp thị và phát triển) về những mong muốn và mục tiêu của họ khi làm ra sản phẩm phần mềm và kèm theo cả những điều mà họ không mong muốn có trong sản phẩm đó. Tài liệu này cũng cung cấp một cơ sở thực tế cho việc ước tính về thời gian thực hiện cũng như chi phí, các rủi ro và lịch trình chi tiết cho việc xây dựng sản phẩm.

2.1.2 Vị trí của SRS trong quá trình xây dựng phần mềm

SRS sẽ được tạo ra ở giai đoạn đầu của một dự án, khi các bên liên quan hình thành ý tưởng và xây dựng yêu cầu cho một sản phẩm phần mềm. Bên phát triển phần mềm cần tổ chức các cuộc họp với các bên liên quan để thu thập yêu cầu, từ đó xây dựng nên các tài liệu đặc tả yêu cầu phần mềm, chính là các SRS.

Các SRS này có thể được coi như một tài liệu chuẩn và được sử dụng xuyên suốt trong suốt quá trình xây dựng phần mềm. Bên sản xuất phần mềm có thể coi như là một bản tài liệu chuẩn đã được thống nhất giữa các bên liên quan, sử dụng cho toàn bộ quá trình coding và testing, cũng như xây dựng các tài liệu đầu ra cho sản phẩm như: Tài liệu hướng dẫn sử dụng, Bộ test case cho Unit test và System test, v.v.

2.1.3 Cấu trúc tổng quan của SRS

Một tài liệu SRS cần bao gồm được toàn bộ nội dung đặc tả yêu cầu mà một sản phẩm phần mềm cần có. Một SRS thông thường cần có các phần như sau:

❖ Giới thiệu chung

- ❖ Yêu cầu tổng quan
- ❖ Yêu cầu chức năng chi tiết
- ❖ Yêu cầu phi chức năng
- ❖ Phụ lục.

2.2 Giới thiệu về Use Case

2.2.1 Khái niệm Use Case

Một Use Case là tất cả những cách thức sử dụng một chức năng hệ thống để đạt được một mục đích nào đó của một người dùng cụ thể. Tập hợp tất cả các Use Case sẽ cho chúng ta một bộ các cách thức hiệu quả để sử dụng hệ thống phần mềm.

Một Use Case là:

- ❖ Một tập hợp tuần tự các hành động mà hệ thống phần mềm thực thi để ra được một kết quả mong muốn cho một người dùng cụ thể.
- ❖ Xác định một hành vi của hệ thống để kết hợp với một người dùng nhằm mục đích tạo ra một giá trị cho người đó trong quá trình sử dụng hệ thống.
- ❖ Là đơn vị nhỏ nhất của các hoạt động cung cấp một kết quả có ý nghĩa cho người dùng.
- ❖ Là một nơi để chứa đựng một bộ các yêu cầu có liên quan đến nhau.

2.2.2 Vai trò của Use Case trong SRS

Trong SRS, một Use Case được trình bày chi tiết trong phần “Yêu cầu chức năng chi tiết”.

2.2.3 Cấu trúc tổng quan của Use Case

Một Use Case sẽ dùng để đặc tả chi tiết một chức năng, và được chia thành các phần chi tiết như sau: Thông tin tổng quan (General Information), Luồng hoạt động (Activities Flow), Các quy tắc nghiệp vụ (Business Rules) và Giả lập màn hình (Mockup Screen).

2.3 Giới thiệu tổng quan về Test Case

2.3.1 Khái niệm Test Case

Test Case, là một tập hợp các điều kiện được coi như một thử nghiệm để xác định liệu một ứng dụng, một hệ thống phần mềm hoặc một trong các tính năng của nó có làm việc như những thiết lập ban đầu hay không.

Các cơ chế để xác định liệu một chương trình phần mềm hoặc hệ thống đã được thông qua một thử nghiệm nay không được biết đến như một quy trình kiểm thử. Có thể cần nhiều trường hợp thử nghiệm để có thể xác định rằng một chương trình phần mềm hoặc hệ thống được coi là xem xét kỹ lưỡng đầy đủ trước khi phát hành hoặc bàn giao sản phẩm.

Hiện nay, tại Việt Nam, ở các công ty sản xuất phần mềm, Test Case hầu hết đều được xây dựng và lưu trữ bằng file excel để thuận tiện cho việc quản lý và chỉnh sửa cũng như bàn giao giữa các bên liên quan. Nội dung của một Test Case có thể như sau:

Req. Id	Test case Id	Test case Title	Test procedure	Expected result	Priority	Test Round 1 Result	Test Round 2 Result	Remarks
Req_3	[FN_121]	Update Applicant Government Agency successfully. - Not change the email.	<u>Step 1:</u> Click <Profile> button at the top right of the INBOX page. <u>Step 2:</u> Update valid data then click <Next> button. <u>Step 3:</u> Input valid captcha then click <Submit> button. <u>Step 4:</u> Click <OK> button.	1. Profile screen is opened. 2. Submit registration form is displayed with captcha generated by the system. 3. Updated Confirmed page is displayed. 4. Update profile successfully and return Home page of Inbox.	High	Fail	Pass	

Table 1: Cấu trúc của một Test Case thông thường.

2.3.2 Vị trí của Test Case trong quá trình xây dựng phần mềm

Test Case được xây dựng ở giai đoạn gần cuối của quy trình sản xuất phần mềm, khi các khâu xây dựng tài liệu SRS, thiết kế và lập trình gần như đã hoàn thiện, các Teste sẽ bắt đầu bắt tay vào xây dựng bộ Test Case dựa trên các yêu cầu nghiệp vụ được mô tả trong tài liệu SRS. Sau đó, các Test Case này sẽ được đưa vào thực thi trong quá trình kiểm thử phần mềm trước khi chương trình phần mềm được đưa vào hoạt động trong thực tế.

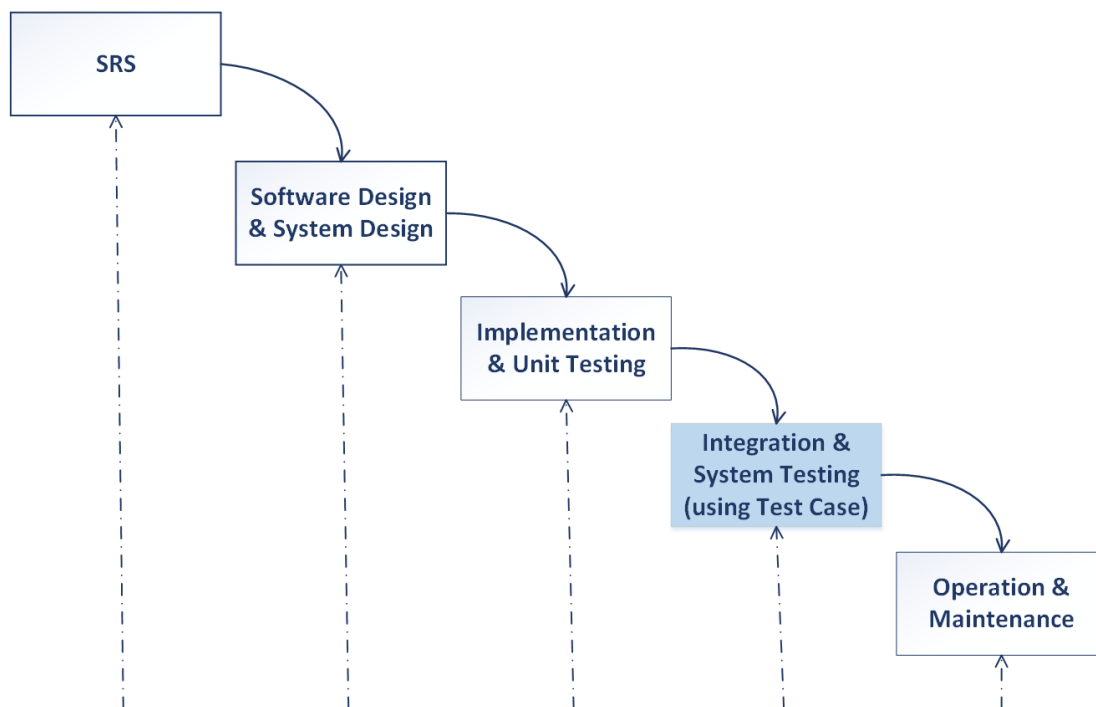


Figure 1: Vị trí của Test Case trong quá trình xây dựng phần mềm

2.3.3 Cấu trúc tổng quan Test Case

Một Test Case có thể bao gồm một bước thực hiện hoặc một bộ các bước thực hiện tuần tự, dùng để kiểm thử tính đúng đắn của hành động/chức năng của một chương trình phần mềm. Các kết quả mong muốn hoặc đầu ra mong muốn của hành động/chức năng luôn phải được đề cập trong một Test Case.

CHƯƠNG 3. GIẢI PHÁP XÂY DỰNG TEST CASE DỰA TRÊN SRS

3.1 Dữ liệu đầu vào

Việc xây dựng bộ Test Case cho một hoặc nhiều chức năng cho một chương trình phần mềm sẽ dựa trên các thông tin được bao gồm trong tài liệu SRS, trong đó các thông tin dùng cho mỗi Test Case sẽ được lấy ra từ các thông tin của Use Case.

Các thông tin của Use Case sẽ được sử dụng để xây dựng Test Case như sau:

3.1.1 Thuộc tính của Use Case

Thuộc tính của một Use Case trong SRS sẽ bao gồm các thông tin như sau:

Objective:	Mục đích hoặc chức năng chính của Use Case.
Actor:	Người thực hiện Use Case.
Trigger:	Hoạt động bắt đầu để thực hiện Use Case.
Pre-condition:	Điều kiện cần thiết để Use Case có thể được thực hiện.
Post-condition:	Kết quả mong đợi sau khi Use Case được thực hiện thành công

Table 2: Thuộc tính của Use Case

3.1.2 Luồng hoạt động (Activity Flow)

Các thông tin trong Activity Flow sẽ được sử dụng để tạo ra nội dung “Các bước thực hiện hoặc thứ tự của hành động” (Test Procedure) trong Test Case.

Trong Activity Flow của Use Case trong SRS, chúng ta sẽ chỉ tập trung vào các bước được thực hiện ở bên phía người dùng (Actor), thay vì các bước thực hiện bên phía hệ thống (system). Các thông tin thực hiện bên phía hệ thống sẽ có thể được sử dụng để làm nội dung cho phần “Kết quả mong đợi” trong Test Case.

Các thông tin trong luồng hoạt động của Use Case có thể được sử dụng để xây dựng Các bước thực hiện (Test Procedure) và Kết quả mong đợi (Expected Result).

3.1.3 Các quy tắc nghiệp vụ (Business Rules)

Các quy tắc nghiệp vụ (Business Rules) là thành phần chính của một Use Case. Phần này sẽ quy định các điều kiện hiển thị, hoạt động cũng như cách thức hoạt động của một Use Case.

Thông thường, một Use Case có thể bao gồm các loại quy tắc nghiệp vụ chính như sau:

Trong luận văn này, tác giả sẽ đề cập tới cách sử dụng từng loại quy tắc nghiệp vụ để xây dựng được các Test Case trong bộ Test Case cho một chương trình phần mềm.

3.2 Dữ liệu đầu ra

Từ thông tin của Use Case được mô tả trong SRS, tác giả của luận văn mong muốn có thể xây dựng một bộ Test Case sử dụng các thông tin đó với các thông tin chi tiết như sau:



3.3 Phương pháp thực hiện

Trong một bộ Test Case, chúng tôi mong muốn có thể xây dựng các Test Case cho từng Use Case riêng biệt, và sẽ gộp các Test Case trong cùng một Use Case thành một nhóm. Như vậy, từ một Use Case, chúng ta có thể xây dựng nên một bộ Test Case cho một Use Case cụ thể theo các quy luật.

CHƯƠNG 4. CÔNG NGHỆ SỬ DỤNG

Như đã trình bày ở trên, trong luận văn này, tôi đã sử dụng một bộ mã nguồn mở POI trong việc đọc và ghi dữ liệu cho Microsoft Word và Microsoft Excel và ngôn ngữ lập trình Java.

4.1 POI Apache

Apache POI là một thư viện mã nguồn mở Java, được cung cấp bởi Apache, nó là một thư viện đầy sức mạnh giúp bạn làm việc với các tài liệu của Microsoft như Word, Excel, Power point, Visio,...

POI là viết tắt của "Poor Obfuscation Implementation". Các định dạng file của Microsoft được giấu kín. Những kỹ sư của Apache phải cố gắng để tìm hiểu nó, và họ thấy rằng Microsoft đã tạo ra các định dạng phức tạp một cách không cần thiết.

4.1.1 Tính năng của Apache POI

Apache POI hỗ trợ bạn làm việc với các định dạng của Microsoft, các class của nó thường có tiếp đầu ngữ HSSF, XSSF, HPSF, ... Nhìn vào tiếp đầu ngữ của một class, bạn có thể biết được class đó hỗ trợ loại định dạng nào.

STT	Tiếp đầu ngữ	Mô tả
1	HSSF (Horrible SpreadSheet Format)	Đọc và ghi file định dạng Microsoft Excel (XLS).
2	XSSF (XML SpreadSheet Format)	Đọc và ghi định dạng file Open Office XML (XLSX).
3	HPSF (Horrible Property Set Format)	Đọc thông tin tóm tắt về tài liệu từ các file Microsoft Office.
4	HWPF (Horrible Word Processor Format)	Mục đích đọc và ghi các file định dạng Microsoft Word 97 (DOC).
5	HSLF (Horrible Slide Layout Format)	Một thực hiện thuần Java cho các file Microsoft PowerPoint.
6	HDGF (Horrible DiaGram Format)	Các thực hiện (implementation) thuần Java khởi đầu cho các file nhị phân Microsoft Visio.
7	HPBF (Horrible PuBlisher Format)	Một thực hiện thuần Java cho các file Microsoft Publisher.

8	HSMF (Horrible Stupid Mail Format)	Một thực hiện thuần Java cho các file Microsoft Outlook MSG
9	DDF (Dreadful Drawing Format)	Một package cho giải mã các định dạng Microsoft Office Drawing.

Apache POI được áp dụng rộng rãi trong các công việc thực tế như sau:

4.1.2 Sử dụng Apache POI trong đọc file SRS

Đọc từ file doc ra dữ liệu XWPFParagraph.

```
// Đọc file
FileInputStream fis = new FileInputStream(input);
XWPFDocument xdoc = new XWPFDocument(OPCPackage.open(fis));
XWPFStyles styles = xdoc.getStyles();
List<XWPFParagraph> paragraphList = xdoc.getParagraphs();
```

Nếu file đọc được là heading thì sẽ kiểm tra xem có phải heading có value là “USE CASE” không.

```
if (style != null) {
    //Nếu style là heading
    if (style.getName().startsWith(Constant.HEADING)) {
        // Đọc heading
        if (style.getName().equals(Constant.HEADING1))
            if (paragraph.getText().equals(Constant.HEADING_UC))
                headingUC1 = paragraph.getText();
            else headingUC1 = "";
        if (headingUC1.equals(Constant.HEADING_UC) && style.getName().equals(Constant.HEADING2)) {
            System.out.println(paragraph.getText());

            UCEntity en = readTableUC(UCCount, input);
            listUC.add(en);
            UCCount++;
        }
    }
}
```

Nếu header là USE CASES thì tiến hành đọc danh sách các table của USE CASE

```
XWPFDocument xdoc=new XWPFDocument(OPCPackage.open(fis));
Iterator<IBodyElement> bodyElementIterator = xdoc.getBodyElementsIterator();
```

Nếu table là USE CASE thì tiến hành đọc Precondition, actor, objective tương ứng với usecase đó.

```

if (table.getRow(0).getCell(0).getText().equals(Constant.USECASE)
    && j==i) {
    System.out.println("precondition : "+ table.getRow(4).getCell(1).getText());
    System.out.println("actor : "+ table.getRow(2).getCell(1).getText());
    System.out.println("objecttive : "+ table.getRow(1).getCell(1).getText());
    uc.setPrecondition(table.getRow(4).getCell(1).getText());
    uc.setActor(table.getRow(2).getCell(1).getText());
    uc.setObjecttive(table.getRow(1).getCell(1).getText());
}

```

Ta có đầu vào của 1 USE CASE

USE CASE	
Objective:	Allows actor to view list of Kneading Command in the system.
Actor:	Terminal A14; Terminal A15.
Trigger:	Clicks on button/link to go to "Create <u>Tabletising Command</u> " page.
Pre-condition:	User logs into the system as above actor successfully.
Post-condition:	The list of Kneading Command is displayed.

Đọc tiếp bảng RULES sau bảng USE CASE

```

XWPFTable table3 = tableList.get(k);
String rules="";
System.out.println("RULES");
for(int x =1;x < table3.getNumberOfRows();x++) {
    List<XWPFParagraph> listpa = table3.getRow(x).getCell(2).getParagraphs();
    for(XWPFParagraph xwp : listpa)
        //System.out.println(table3.getRow(x).getCell(2).getParagraphs().get(0).getParagraphText());
        if(xwp.getNumLevelText() != null) System.out.println(xwp.getNumLevelText().hashCode() + " : " +xwp.get
}

```

Ta có đầu vào của 1 RULES

RULES		
Step	BR Code	Description
(2)	BR 1	<p><u>Displaying rules:</u></p> <ul style="list-style-type: none"> ❖ The system will display the screen as Screen 1. ❖ The system displays the list of Kneading Command that retrieved from "Input Kneading Command" form where [Status] = "Yet" or [Status] = "Command". <ul style="list-style-type: none"> ➢ Aaaaa ➢ Bbbb ➢ ❖ Sorting rules: N/A – means the system will not sort the item list. ❖ No filter. ❖ Paging rules: Please refer to section "Paging" on the KCSG_Overall Requirement_SRS_v0.5 document.

4.2 JXLS

4.2.1 Giới thiệu

- ❖ Jxls là một thư viện Java hỗ trợ cho việc xuất file excel. Jxls sử dụng một đánh dấu đặc biệt trong Excel mẫu để xác định định dạng đầu ra và bố trí dữ liệu.
- ❖ Java đã có rất nhiều mã nguồn mở và các thư viện thương mại để tạo tập tin Excel (của những mã nguồn mở đáng nói là Apache POI và Java Excel API).
- ❖ Những thư viện khác sẽ phải sử dụng rất nhiều câu lệnh Java để tạo ra 1 file Excel đơn giản.

4.2.2 Tính năng, đặc điểm

Các tính năng và đặc điểm nổi bật của JXLS:

- ❖ Định dạng Excel đầu ra là XML và nhị phân (phụ thuộc vào mức độ phát triển)
- ❖ Tạo thành 1 tập hợp các dòng và cột
- ❖ Xây dựng đầu ra có điều kiện
- ❖ Đánh dấu được ngôn ngữ biểu thức
- ❖ Tạo nhiều sheet output
- ❖ Dùng trực tiếp các công thức của Excel
- ❖ Sử dụng các công thức kèm tham số
- ❖ Hỗ trợ việc Merge Cell
- ❖ Sử dụng các comments trong markup của Excel để định nghĩa công thức
- ❖ Tùy chỉnh các công thức

4.2.3 Sử dụng JXLS để tạo file excel

Ta có định dạng file Excel để sử dụng làm template output:

UC Id	Test case Id	Test case Title	Test procedure	Expected result	Priority	Test Round 1 Result	Test Round 2 Result	Remarks
UC 1: Update Applicant User								
		- Pre-condition: \${uc.precondition} - Actor: \${uc.actor}						
UC_1	[FN_121]	\${uc.objective}	\${uc.actorStep}	\${uc.systemStep}	High	Fail	Pass	

Sử dụng các comment trong excel để định nghĩa dữ liệu truyền vào:

UC Id	Test case Id	Test case Title
UC 1: Update Applicant Lear	7Love:	
	[-	eco
UC_1	[F	ve}

- Items: danh sách dữ liệu sẽ được in ra output.
- lastCell: cell cuối cùng trong template output của 1 dữ liệu.

Thực hiện đọc file template (output) và ghi ra file Result.xlsx

```
try(InputStream is = new FileInputStream(output)) {
    try (OutputStream os = new FileOutputStream("D:/doanws/result.xlsx")) {
        Context context = new Context();
        int i=0;
        for(UCEntity uc : listEntity) {
            uc.setUcid("UC_" + (++i));
            uc.setFnid("FN_" + i);
        }
        context.putVar("lsuc", listEntity);
        JxlsHelper.getInstance().processTemplate(is, os, context);
    }
}
```

Hiện tại, chúng tôi sử dụng bộ tài liệu SRS của Applicant Registration Module (Đăng ký người dùng) của dự án phần mềm tại công ty FPT Software khi làm việc với khách hàng Singapore để làm ví dụ minh họa cho việc sinh test cases.

- File đầu vào của chương trình là một tài liệu đặc tả yêu cầu nghiệp vụ như sau:



- Khi sử dụng chương trình, đã tạo ra được bộ Test Case với các test cases như trong file dưới đây.



KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong luận văn này, tác giả đã hoàn thành được những công việc cụ thể sau:

- ❖ Hoàn thiện việc nghiên các khái niệm và chức năng liên quan của SRS và Test Case.
- ❖ Xây dựng các công thức sinh Test Case từ các yêu cầu nghiệp vụ.
- ❖ Xây dựng chương trình sinh tự động bộ Test Case từ một tài liệu đặc tả yêu cầu nghiệp vụ SRS.

Hiện nay, chương trình đã có thể giải quyết các bài toán với các điều kiện như sau:

- ❖ Các đặc tả yêu cầu nghiệp vụ được trình bày rõ ràng theo khung của từng Use Case như đã đề cập trong phần Giới thiệu về Use Case.
- ❖ Các đặc tả yêu cầu nghiệp vụ của mỗi Use Case được chia thành các Business Rules như sau:
 - Displaying/Showing rules.
 - Validation rules.
 - Updating rules.
 - Saving rules.
 - Deleting rules.

Định hướng tương lai tác giả mong muốn có thể phát triển đề tài để có thể tự động sinh bộ Test Case cho các chức năng hệ thống phức tạp hơn như các Batch Job hoặc các Timer Job của hệ thống, các chức năng chạy không cần tác động của user, hoặc các chức năng trả về những kết quả trừu tượng.

TÀI LIỆU THAM KHẢO

1. Glenford J. Myers, Tom Badgett and Corey Sandler (2015), *The Art of Software Testing, Third Edition*.
2. *Practice Book for the Paper-based GRE ® revised General Test (PDF), Second Edition*.
3. Glenn Fulcher and Fred Davidson (2006), *Language Testing and Assessment: An Advanced Resource Book*.
4. Rekard Edgren (2012), *The Little Black Book on Test Design*.
5. Cem Kaner and Jack Falk, *Testing Computer Software*.
6. IIBA | International Institute of Business Analysis (2015), *A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide), Third Edition*.
7. IEEE Computer Society (1998), *IEEE Recommended Practice for Software Requirements Specifications*.
8. John D. Gannon, James M. Purtilo, Marvin V. Zelkowitz (2001), *MarylandSOFTWARE SPECIFICATION: A Comparison of Formal Methods*, Department of Computer Science University of Maryland College Park

PHỤ LỤC

Phụ lục 4: THÔNG TIN LUẬN VĂN THẠC SĨ

ĐẠI HỌC QUỐC GIA HÀ NỘI

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Độc lập - Tự do - Hạnh phúc

THÔNG TIN VỀ LUẬN VĂN THẠC SĨ

1. Họ và tên học viên: Bùi Thị Thúy
2. Giới tính: Nữ
3. Ngày sinh: 13/02/1991
4. Nơi sinh: Việt Thuận, Vũ Thư, Thái Bình.
5. Quyết định công nhận học viên số:, ngày.....tháng.....năm.....
6. Các thay đổi trong quá trình đào tạo: Không
7. Tên đề tài luận van:
8. Chuyên ngành: Hệ thống thông tin
9. Mã số:
10. Cán bộ hướng dẫn khoa học: ThS. Trương Ninh Thuận
11. Tóm tắt các kết quả của luận van:
*nêu tóm tắt các kết quả của luận van, nhấn mạnh các **kết quả mới** nếu có)*
12. Khả năng ứng dụng trong thực tiễn: (nếu có)
13. Những hướng nghiên cứu tiếp theo: (nếu có)
14. Các công trình đã công bố có liên quan đến luận van:
liệt kê các công trình theo thứ tự thời gian nếu có)

Ngày tháng năm 20

Xác nhận của cán bộ hướng dẫn

(Kí và ghi rõ họ tên)

Ngày tháng năm 20

Học viên

(Kí và ghi rõ họ tên)