

การทดสอบฟีเจอร์สำหรับควบคุมการใช้งานบัตรเครดิตบน
แอปพลิเคชันยูซูส
UCHOOSE APPLICATION FEATURE TESTING FOR
SPENDING CONTROL

เกษราภรณ์ โคเฟื่อง

สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565



U CHOOSE APPLICATION FEATURE TESTING FOR SPENDING CONTROL

KETSARAPORN KOFEUNG

A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2022

หัวข้อสหกิจศึกษา	การทดสอบฟีเจอร์สำหรับควบคุมการใช้งานบัตรเครดิตบนแอปพลิเคชันยูซูส
	UCHOOSE APPLICATION FEATURE TESTING FOR SPENDING CONTROL
ชื่อนักศึกษา	นางสาวเกษราภรณ์ โคเฟื่อง 62050130
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ดร.อัคเดช อุดมชัยพร

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต ประจำปีการศึกษา 2565

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.วรางคณา กิมปาน ประธานกรรมการ	
ดร.อัคเดช อุดมชัยพร กรรมการและอาจารย์ที่ปรึกษา	

ลิขสิทธิ์ของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อสหกิจศึกษา	การทดสอบพีเจียร์สำหรับควบคุมการใช้งานบัตรเครดิตบนแอปพลิเคชันยูซูส
ชื่อนักศึกษา	นางสาวเกษราภรณ์ โคเฟื่อง 62050130
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ดร.อัคเดช อุดมชัยพร

บทคัดย่อ

สหกิจศึกษานี้มีวัตถุประสงค์เพื่อศึกษาการทดสอบ โดยมีการทดสอบ APIs (Application Programming Interfaces) และการทดสอบส่วนต่อประสานกับผู้ใช้ (UI) ของพีเจียร์ Spending Control บนแอปพลิเคชันยูซูส ตั้งแต่ต้นจนจบไฟล์ โดยมีการทดสอบทั้งแบบ Manual Testing ซึ่งเป็นการทดสอบส่วนต่อประสานกับผู้ใช้ และ APIs บางส่วน และ Automation Testing เป็นการสอบของ API โดยใช้ Postman เป็นเครื่องมือสำหรับทดสอบ และได้มีการเขียน Test Script สำหรับการทดสอบ APIs แบบ Automation

คำสำคัญ : Application Programming Interface, UI, Spending Control, UCHOOSE, Manual Testing, Automate Testing, Postman

Title	UCHOOSE APPLICATION FEATURE TESTING FOR SPENDING CONTROL
Students	Ketsaraporn Kofeung
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
School	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2022
Advisor	Dr. Akadej Udomchaiporn

Abstract

The purpose of this cooperative education program is to learn how to test software with APIs (Application Programming Interfaces) and user interface (UI) testing for the Spending Control feature on the UCHOOSE application. There is both manual testing, which tests user interfaces and some APIs, and automation testing, which test APIs using Postman as a testing tool and create a test script for testing the automation APIs.

Keywords : Application Programming Interface, UI, Spending Control, UCHOOSE, Manual Testing, Automate Testing, Postman

กิตติกรรมประกาศ

สหกิจศึกษาครั้งนี้เสร็จสมบูรณ์ได้ ต้องขอขอบคุณ คุณแม่ เพื่อน ๆ และอาจารย์ที่ปรึกษาทุกท่านสำหรับทุกคำปรึกษา แนวคิด คำแนะนำต่าง ๆ และกำลังใจที่ดีเสมอมา

ขอบคุณเกมแกนชินอิมแพกต์ สำหรับเนื้อเรื่อง ตัวละคร และอีเอนท์ที่สนุกสนาน และช่วยผ่อนคลายในช่วงระยะเวลาที่ตึงเครียดได้เป็นอย่างดี

และสุดท้าย ขอขอบคุณ คุณกวียา พรรณรัตน์พงศ์ คุณจตุพล วาริบุญมา และ คุณภาวิณี ไชยสถิตย์ พี่เลี้ยงและเมนเจอร์สุดน่ารัก และแสนใจดีที่มอบความรู้ และประสบการณ์การทำงาน อีกทั้งยังคอยช่วยเหลือในครั้งที่ข้าพเจ้าพบเจอปัญหา ทั้งปัญหาในการทำงาน และปัญหาอื่น ๆ อีกทั้งยังมีประสบการณ์การทำงานของพี่ ๆ มาเล่าให้ฟังอยู่เสมอ ขอขอบคุณความน่ารักของพี่ ๆ ที่ทำให้ประสบการณ์การทำงานครั้งแรกเป็นประสบการณ์ที่น่าจดจำและไม่น่ากลัว

การมาสหกิจศึกษาครั้งนี้จะไม่มี ความหมาย หากขาดคำแนะนำ และกำลังใจจากทุก ๆ ท่าน จึงขอกราบขอบพระคุณเป็นอย่างสูง

เกษราภรณ์ โคเฟื่อง

สารบัญ

บทคัดย่อ.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ภาษาโปรแกรมมิ่งที่ใช้ในการทดสอบซอฟต์แวร์.....	4
2.1.1 JavaScript Language	4
2.1.2 Structured Query Language (SQL)	4
2.2 เครื่องมือที่ใช้ในการทดสอบ	5
2.2.1 TestRail.....	5
2.2.2 Postman.....	6
2.2.3 Proxyman.....	8
2.2.4 SQLDeveloper.....	9
2.3 กระบวนการและเครื่องมือที่ใช้ในการทำงาน	9
2.3.1 Agile	9
2.3.2 Jira	11
2.3.3 Miro	12
2.4 ประเภทและระดับของการทดสอบ	12
2.4.1 Manual Testing.....	12

2.4.2 Fuctional Testing.....	13
2.4.3 Automation Testing.....	13
2.4.4 System integration Testing	14
2.4.5 User Acceptance Testing	14
บทที่ 3 วิธีการดำเนินงานวิจัย.....	16
3.1 ตารางการปฏิบัติงาน	17
3.2 Use Case ของพีเจอร์ Spending Control	19
3.3 การดำเนินงาน.....	23
3.3.1 สร้าง Test Case	23
3.3.2 ทำการทดสอบ	24
3.3.3 เก็บผล	24
บทที่ 4 ผลการวิจัยและการอภิปรายผล	25
4.1 การทดสอบด้วย Manual Testing	25
4.2 การทดสอบด้วย Automation Testing	42
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ	48
5.1 สรุปผลการวิจัย	48
5.2 ข้อเสนอแนะ.....	49
เอกสารอ้างอิง.....	50
ภาคผนวก.....	52

สารบัญตาราง

ตารางที่	หน้า
3.1 ตารางการปฏิบัติงาน เดือนมิถุนายน 2565 ถึง ธันวาคม 2564.....	17
3.2 ตารางคำอธิบาย Use Case Register Application	20
3.3 ตารางคำอธิบาย Use Case View quickView Screen	21
3.4 ตารางคำอธิบาย Use Case Receive Value	21
3.5 ตารางคำอธิบาย Use Case Verify Eligible Card	22
3.6 ตารางคำอธิบาย Use Case Edit Card Control.....	23
4.1 Test Case สำหรับใช้ในการทดสอบในส่วนของ “การจำกัดวงเงิน” หรือ “LIMIT CARD USAGE”	25
4.2 Test Case สำหรับใช้ในการทดสอบในส่วนของ “การแก้ไขรายการควบคุมการใช้จ่าย”	28
4.3 Test Case สำหรับใช้ในการทดสอบในส่วนของ “การบันทึกรายการควบคุมการใช้จ่าย”	34

สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างของ TestRail.....	5
2.2 Postman.....	6
2.3 Environment	7
2.4 Send Request	8
2.5 Test Result	8
3.1 Use Case Diagram.....	20

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ธุรกิจบัตรเครดิตในปัจจุบัน มีแนวโน้มการเติบโตอย่างต่อเนื่องเนื่องจากความสะดวกสบายในไลฟ์สไตล์ต่าง ๆ ของผู้บริโภค ทั้งการใช้บัตรเพื่อเบิกถอนเงินสดล่วงหน้า การใช้จ่ายผ่านคิวอาร์โค้ด ตลอดจนการชำระเงินกับทางร้านค้าด้วยบัตรโดยตรงทั้งในประเทศและต่างประเทศ หรือการชำระเงินจากทางเว็บไซต์ ซึ่งล้วนแต่ทำได้ง่ายและสะดวกสบายโดยการใช้เพียงหมายเลขบนบัตรเครดิตของผู้ใช้งาน วันหมดอายุ และเลข CCV (Card Verification Value Code) ซึ่งวิธีการดังกล่าวนี้ หากทางเว็บไซต์มีการรักษาความปลอดภัยที่ไม่ดีพออาจจะทำให้ข้อมูลในส่วนนี้หลุดไปถึงอาชญากรและก่อให้เกิดความเสียหายได้ ทางกรุงศรี คอนซูมเมอร์จึงได้จัดทำฟีเจอร์ “Spending Control” บนแอปพลิเคชัน UCHOOSE เพื่อให้ผู้ใช้งานสามารถควบคุมการใช้งานบัตรเครดิตได้ด้วยตนเอง

ฟีเจอร์สำหรับควบคุมการใช้งานบัตรเครดิต หรือ Spending Control นั้น ผู้ใช้งานจะสามารถควบคุมการใช้งานของบัตรเครดิตของตนเอง โดยการใช้จ่ายโดยตรงผ่านหน้าร้าน และการใช้จ่ายออนไลน์ ผู้ใช้งานจะสามารถกำหนดวงเงินการใช้จ่ายต่อวัน กำหนดวงเงินการใช้จ่ายต่อรายการ และจำนวนรายการใช้จ่ายต่อวันได้ ส่วนการกดเงินจากตู้ ATM นอกจากการกำหนดวงเงินในการเบิกเงินสดล่วงหน้าแล้ว ผู้ใช้งานยังสามารถเปิดหรือปิดการใช้งานการกดเงินจากตู้ ATM ได้ ซึ่งการใช้จ่ายผ่านคิวอาร์โค้ดผู้ใช้งานก็สามารถเปิดหรือปิดการใช้งานได้เช่นเดียวกัน โดยการควบคุมการใช้งานบัตรเครดิตนี้ ผู้ใช้งานสามารถปรับเปลี่ยนได้ไม่จำกัดจำนวนครั้งในแต่ละวัน แต่จะไม่สามารถทำรายการได้ในเวลา 22:00 น. - 1:00 น. ของทุกวัน

ดังนั้น นอกจากการพัฒนาฟีเจอร์สำหรับควบคุมการใช้งานบัตรเครดิตแล้ว ต้องมีการทดสอบและตรวจสอบคุณภาพ เพื่อความถูกต้องและสมบูรณ์ โดยทางผู้จัดทำได้มีส่วนร่วมในการทดสอบและตรวจสอบคุณภาพทั้งในส่วนของ API และ Client ตั้งแต่ต้นจนจบการทำงานของฟีเจอร์ดังกล่าว

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อศึกษาการทำการทดสอบประเภทต่าง ๆ เช่นการทดสอบ API หรือการทดสอบบน Client อย่างสมาร์ตโฟนบนระบบปฏิบัติการ Android และ iOS
- 2) เพื่อศึกษาการทำ Automation Testing
- 3) เพื่อศึกษาเครื่องมือต่าง ๆ ที่ใช้ในการทดสอบ
- 4) เพื่อตรวจสอบความถูกต้องก่อนทำการปล่อยตัวผลิตภัณฑ์ออกสู่ตลาด

1.3 ขอบเขตของงานวิจัย

ด้านการทดสอบซอฟต์แวร์

1) สามารถออกแบบและวางแผนแอปพลิเคชันในฐานะสมาชิกทีมพัฒนาซอฟต์แวร์ ด้วยการทำงานแบบ Agile

2) สามารถทดสอบซอฟต์แวร์ได้อย่างมีคุณภาพและตรงตามมาตรฐาน

3) สามารถสร้าง Automated Test Frameworks (Web, APIs, Mobile)

ด้านระยะเวลา

เริ่มตั้งแต่ 15 มิถุนายน 2565 ถึง 31 ธันวาคม 2565

ด้านอุปกรณ์

ด้านฮาร์ดแวร์ (Hardware)

1) MacBook Pro (15-inch, 2018)

Processor : 2.2 GHz 6-Core Intel Core i7

Memory : 16 GB 2400 MHz DDR4

Storage : 251 GB Flash Storage

OS : macOS

2) DELL Latitude E4740

Processor : Intel Core i5

Memory : 8 GB DDR4

Storage : 256 GB

OS : Windows 10

3) iPhone 13 mini

Chipset : Apple A15 Bionic

Storage : 128 GB

OS : iOS 15.5

4) HUAWEI Y7a

RAM : 4 GB

ROM : 128 GB

OS : EMUI 10.1.1 /Android 10

ด้านซอฟต์แวร์ (Software)

1) Postman

2) Swagger

3) Jira

4) Microsoft team

- 5) Gitlab
- 6) Slack
- 7) Portainer
- 8) Proxyman
- 9) TestRail
- 10) Miro
- 11) Figma
- 12) TablePlus
- 13) Sourcetree
- 14) SQLDeveloper

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เข้าใจกระบวนการทำงานของ QA สามารถวิเคราะห์ และออกแบบสถานการณ์สำหรับการทดสอบ ตลอดจนการส่งมอบงานอย่างตรงตามกำหนด
- 2) สามารถทำออกแบบ Test Script สำหรับใช้ในการทดสอบแบบอัตโนมัติ หรือ Automated Test ได้
- 3) มีทักษะในการทำงานเป็นทีมและการสื่อสารที่ดีขึ้น สามารถแก้ไขปัญหาเฉพาะหน้า และวางแผนการทำงานและส่งมอบงานให้ทันระยะเวลาภายในกำหนด

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการทดสอบฟีเจอร์ “Spending Control” บนแอปพลิเคชัน UCHOOSE ผู้จัดทำได้ค้นคว้าเอกสารที่เกี่ยวข้องเพื่อใช้สำหรับการทดสอบ โดยแบ่งเป็นหัวข้อได้ทั้งหมด 4 หัวข้อ ดังนี้

2.1 ภาษาโปรแกรมมิ่งที่ใช้ในการทดสอบซอฟต์แวร์

ในการทดสอบแอปพลิเคชัน UCHOOSE นั้น ได้ใช้ภาษาโปรแกรมมิ่ง 2 ภาษามาใช้ร่วมกับการทดสอบ โดยภาษาแรกคือ ภาษา JavaScript โดยใช้เพื่อสร้าง Script สำหรับการทดสอบ API และภาษา SQL ใช้สำหรับการเรียกดูข้อมูลภายในฐานข้อมูล โดยรายละเอียดของแต่ละภาษามีดังนี้

2.1.1 JavaScript Language

ภาษา JavaScript หรือย่อ JS เป็นภาษาเขียนโปรแกรมที่ถูกพัฒนาและปฏิบัติตามข้อกำหนดมาตรฐานของ ECMAScript ภาษา JavaScript นั้นเป็นภาษาระดับสูง คอมไพล์ในขณะที่โปรแกรมรัน (JIT) และเป็นภาษาเขียนโปรแกรมแบบหลายกระบวนทัศน์ เช่น การเขียนโปรแกรมเชิงขั้นตอน การเขียนโปรแกรมเชิงวัตถุ หรือการเขียนโปรแกรมแบบ Functional ภาษา JavaScript มีไวยากรณ์ที่เหมือนกับภาษา C ใช้วงเล็บเพื่อกำหนดบล็อกของคำสั่ง นอกจากนี้ JavaScript ยังเป็นภาษาที่มีประเภทข้อมูลแบบไดนามิกส์ เป็นภาษาแบบ Prototype-based และ First-class function

ภาษา JavaScript นั้นถือว่าเป็นเทคโนโลยีหลักของการพัฒนาเว็บไซต์ (World Wide Web) ซึ่งทำให้หน้าเว็บสามารถตอบโต้กับผู้ใช้ได้โดยไม่ต้องรีเฟรชหน้าใหม่ (Dynamic Website) เว็บไซต์จำนวนมากใช้ภาษา JavaScript สำหรับควบคุมการทำงานที่ฝั่ง Client-side ซึ่งทำให้เว็บเบราว์เซอร์ต่างๆ มี JavaScript engine ที่ใช้สำหรับประมวลผลสคริปของภาษา JavaScript ที่รันบนเว็บเบราว์เซอร์

2.1.2 Structured Query Language (SQL)

SQL หรือ Structured Query Language เป็นภาษาที่ใช้สำหรับบริหารและจัดการฐานข้อมูล เช่นการเพิ่ม แก้ไข หรือลบข้อมูลที่เราไม่ต้องการ ไปจนถึงการดึงข้อมูลออกมาใช้งาน โดยตัวข้อมูลจะถูกเก็บในรูปแบบของตารางที่มีลักษณะเป็นแถวตอนและแถวตอน โดยข้อมูลที่อยู่ในแถวเดียวกันคือข้อมูลในชุดเดียวกัน ตัวอย่างภาษา SQL ที่ใช้สำหรับการจัดการฐานข้อมูลได้แก่

```
SELECT column1, column2, ...  
FROM table_name;
```

ตัวอย่างคำสั่งข้างต้น เป็นการเลือกแถวตอนที่เรต้องการ ซึ่งหากต้องการเลือกทุกแถวตอนที่มีในตาราง สามารถใช้สัญลักษณ์ * (Star) แทนการพิมพ์ทุกแถวตอนที่มีในตารางได้ และใช้คำสั่ง FROM ในการชี้ว่าข้อมูลที่เราสนใจนั้นอยู่ที่ตารางอะไร

WHERE condition;

สำหรับคำสั่งข้างต้น เป็นการใส่เงื่อนไขที่ต้องการโดยใช้คำสั่ง WHERE โดยคำสั่ง WHERE นั้น ไม่ได้จำกัดว่าต้องใช้ร่วมกับคำสั่ง SELECE ยังสามารถใช้ร่วมกับคำสั่ง UPDATE และ DELETE ได้อีกด้วย

2.2 เครื่องมือที่ใช้ในการทดสอบ

2.2.1 TestRail

TestRail เป็น web-based management สำหรับจัดการ Test Case เพื่อช่วยให้ทีมสามารถจัดระเบียบในการทดสอบ รวมไปถึงข้อมูลความคืบหน้าของกระบวนการทดสอบอย่างเรียลไทม์

TestRail สามารถใช้งานได้แม้ทีมมีขนาดเล็กเพียง 1 คน หรือจะทีมใหญ่ถึง 1,000 คน อีกทั้งยังเหมาะกับการทำงานทุกรูปแบบ ไม่ว่าจะเป็น Traditional Testing หรือ Agile Testing และด้วยตัวโครงสร้างของ TestRail ที่ประกอบไปด้วย Preconditions, Steps และ Expected Results ดังรูป 2.1 จึงง่ายและสามารถจัดการกับ Test Case ได้อย่างมีประสิทธิภาพ

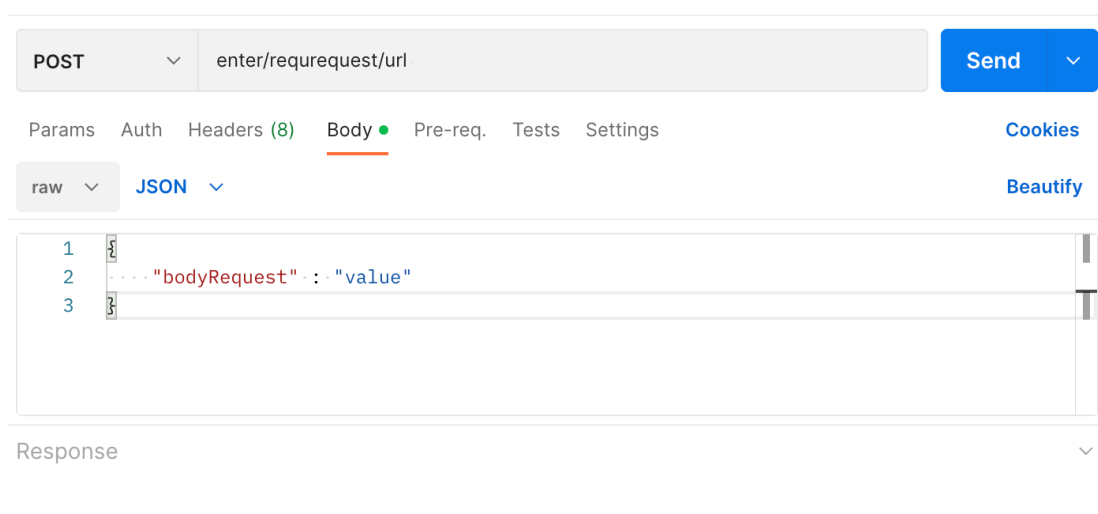
The screenshot shows the 'Add Test Case' form in TestRail. The form is divided into several sections: 'Title' with a text input field; 'Section' (WEC-APP), 'Template' (Test Case (Text)), 'Type' (Other), and 'Priority' (Medium) as dropdown menus; 'Estimate' and 'References' as text input fields; 'Automation Type' as a dropdown menu set to 'None'; 'Preconditions' with a text area and a note 'The preconditions of this test case. Reference other test cases with [C#] (e.g. [C17])'; 'Steps' with a text area and a note 'The required steps to execute the test case.'; and 'Expected Result' with a text area and a note 'The expected result after executing the test case.' At the bottom, there are three buttons: 'Add Test Case' (green), 'Add & Next' (green), and 'Cancel' (red).

รูปที่ 2.1 โครงสร้างของ TestRail

นอกจากการจัดการรายละเอียดของ Test Case แล้ว TestRail ยังสามารถจัดระเบียบของ Test Case ใน Section ต่าง ๆ โดยการเลือก Test Case และยังสามารถอัปเดต และติดตามสถานะของ Test Case นอกจากนี้ TestRail ยังสามารถติดตาม Defect และสามารถทำงานร่วมกับ Atlassian Jira กับ test automation tools เช่น Ranorex Studio เป็นต้น

2.2.2 Postman

Postman เป็น API Platform สำหรับนักพัฒนาในการสร้าง หรือการทดสอบ APIs โดยจะกล่าวถึงในส่วนของการทดสอบ API เป็นหลัก



รูปที่ 2.2 Postman

ด้วย UI ของ Postman ที่ออกแบบมาใช้งานง่ายดังรูป 2.2 ทำให้สามารถส่ง Request ได้โดยเพียงคัดลอก Request URL ที่เราต้องการทดสอบไปวาง อีกทั้งยังสามารถเลือกเมธอดได้อีกมากมาย ไม่ว่าจะเป็น POST, GET, PUT และเมธอดประเภทอื่น ๆ

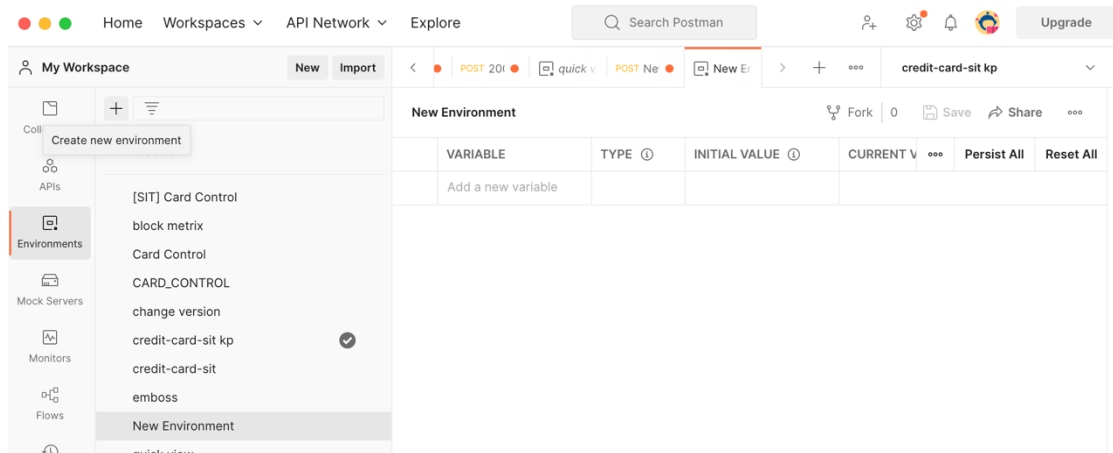
การเขียน Test Script ด้วย Postman

ในการทดสอบ API การตั้งค่า Environment เพื่อเก็บค่าตัวแปรต่าง ๆ สำหรับนำกลับไปใช้อีกครั้ง เพื่อความสะดวก และลดข้อผิดพลาดที่เกิดขึ้นหากเราต้องคอยเปลี่ยนค่าตัวแปรใดตัวแปรหนึ่งหลาย ๆ ครั้ง ซึ่งการตั้งค่า Environment สามารถทำได้ดังนี้

- ผู้ใช้งานสามารถสร้าง Environment ได้โดยไปที่แท็บ Environment ด้านซ้ายมือสุด จากนั้นเลือก Create New Environment โดยผู้ใช้งานสามารถตั้งชื่อ Enviroment ให้

สอดคล้องกับโปรเจก จากนั้นผู้ใช้งานสามารถเพิ่มตัวแปรที่ต้องการได้โดยการพิมพ์ที่

Add a new variable



รูปที่ 2.3 Environment

- ผู้ใช้งานสามารถใช้คำสั่ง `pm.environment.set("variable_name", value)` ได้ที่แท็บ Tests ตามรูป 2.2 โดย value สามารถเป็นได้ทุกค่าไม่ว่าจะเป็นค่าคงที่ ค่าที่รับมาจากไฟล์ CSV หรือจะเป็นค่าที่ดึงมาจาก Response ของ API เส้นดังกล่าว เพื่อนำไปใช้กับ API เส้นถัดไป
- ในการเรียกใช้ตัวแปร ผู้ใช้งานสามารถเรียกใช้ได้โดยการพิมพ์ปีกกา 2 ชั้น และชื่อตัวแปร เช่น `{{variable_name}}`

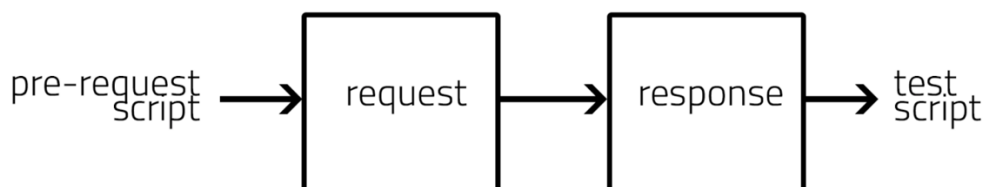
หลังจากการเตรียม Environment เสร็จสิ้น ผู้ใช้งานเตรียม Request ที่ต้องการทดสอบโดยการกลับไปแท็บ API เลือก Request method ตามที่เหมาะสมกับงาน เขียน Pre-request Script ซึ่งเป็นขั้นตอนก่อนที่จะเริ่มยิง Request ที่เราสนใจ เช่น การเขียน Script เพื่อการเก็บ token ซึ่งต้องทำก่อนทุกครั้งก่อนจะยิง Request จากนั้นจึงเริ่มเขียน Test Script ที่แท็บ Tests ซึ่งภาษาที่ใช้ในการเขียน Test Script นั้นคือภาษา JavaScript

ในการทดสอบ API จะมีการทดสอบ Response ซึ่งสามารถกำหนดค่าตัวแปรให้รับมาจากตัว Response ซึ่งเป็นตัว json ได้โดยใช้คำสั่ง `let jsonData = pm.response.json()` โดยตัวอย่างคำสั่งสำหรับการทดสอบมีดังนี้

- `pm.response.to.have.status(200)` เป็นคำสั่งสำหรับตรวจสอบสเตตัสว่าหลังจากยิง Request เส้นนี้แล้ว สเตตัสที่ได้ใช่สเตตัสที่เราต้องการหรือไม่ โดยหากรับกลับมาเป็น 200 แสดงว่า Request ที่ยิงไปนั้นสำเร็จ หรือหากรับกลับมาเป็น 204 แสดงว่าไม่มีข้อมูลสำหรับ Request ที่เราส่งไป เป็นต้น
- `pm.expect(jsonData.key).equal("expected_result")` เป็นคำสั่งสำหรับเปรียบเทียบ response ที่ได้นั้นมีค่าตรงกับที่เราคาดหวังหรือไม่

- `pm.expect(jsonData.key).to.include("expected_result")` เป็นคำสั่งสำหรับเปรียบเทียบ Response ที่ได้นั้นประกอบไปด้วยค่าที่เราต้องการหรือไม่

โดยในการส่ง Request หนึ่งครั้งนั้นจะเริ่มจากการรันตัว Pre-request จากนั้นจึงจะรันตัว Request และได้รับ Response ตามลำดับ และสุดท้ายตัว Test Script จะถูกรับเพื่อทำการทดสอบ ดังรูป 2.4



รูปที่ 2.4 Send Request

โดยหลังจากทำการส่ง Request และทำการทดสอบเสร็จสิ้น สามารถดูผลทดสอบได้โดยไปที่แท็บ Test Result ด้านล่างดังรูป 2.5



รูปที่ 2.5 Test Result

2.2.3 Proxyman

Proxyman คือเครื่องมือที่เอาไว้สำหรับ ดักจับ, ตรวจสอบ รวมไปถึงการแก้ไข HTTP Request ที่เกิดขึ้นใน Network โดยผู้จัดทำได้ใช้ Proxyman รวมกับการทดสอบ Client เพื่อดูพฤติกรรมของแอปพลิเคชัน UCHOOSE รวมถึงการใช้งานพีเจอร์ Map Local หรือก็คือการแก้ไข HTTP status ที่ควรรับมาจาก Server เป็นค่าที่เราต้องการ เช่น 204 No Content ในกรณีที่ต้องการทดสอบพฤติกรรมของตัวแอปพลิเคชันว่าตอบสนองได้ตรงตาม Requirement หรือไม่

2.2.4 SQLDeveloper

SQLDeveloper คือเครื่องมือสำหรับจัดการฐานข้อมูล เช่น การจัดการ User, การสร้าง Table, View, Trigger, Stored Procedure, การ Export/Import ตัว Oracle SQL Developer ถูกพัฒนาด้วยภาษา Java ก่อนการใช้งานจะต้องทำการติดตั้งตัว Java Runtime ให้เรียบร้อย

2.3 กระบวนการและเครื่องมือที่ใช้ในการทำงาน

2.3.1 Agile

Agile Methodology คือแนวคิดในการทำงานที่ไม่ได้จำกัดว่าเป็นแนวคิดสำหรับการพัฒนาผลิตภัณฑ์ในสายซอฟต์แวร์เท่านั้น โดย Agile ให้ความสำคัญในการสื่อสารกับผู้เกี่ยวข้องทุกฝ่าย และการปรับปรุงพัฒนาผลิตภัณฑ์อยู่ตลอดเวลา เพื่อตอบสนองผู้ใช้งาน

ตำแหน่งหน้าที่ในทีมพัฒนาผลิตภัณฑ์แบบ Agile

Stakeholders

- ผู้มีส่วนเกี่ยวข้องกับผลิตภัณฑ์ ผู้ใช้งาน (End user), ผู้บริหารของบริษัท, บริษัทคู่สัญญาที่มาจ้างงานบริษัท โดย Stakeholders เป็นผู้ให้ Requirement

Product Owner

- เป็นตำแหน่งสำหรับทำบางอย่างเพื่อแก้ไขปัญหาหรือตอบสนองความต้องการของ Stakeholders
- เปลี่ยน Requirement เป็น User Story และ Acceptance Criteria รวมถึงการแปลง User Story ที่มีขนาดใหญ่ให้มีขนาดเล็กลง ชัดเจนมากขึ้น และทำให้ทุกฝ่ายเห็นภาพของ User Story ตรงกัน
- จัดลำดับความสำคัญของงานโดยคำนึงถึง User Impact, Business Impact และ Development Cost สำหรับ User Impact และ Business Impact สามารถรู้ได้จากการทำวิจัย ส่วน Development Cost นั้นต้องถามทีมพัฒนาว่าพัฒนายากหรือง่าย ใช้เวลาในการพัฒนาเท่าไร
- ต้องปฏิเสธ Requirement บางอย่าง ไม่ปล่อยให้ Backlog ค้างมากเกินไปและรักษาสมดุลระหว่างการพัฒนาฟีเจอร์ใหม่ การแก้ไขปัญหาเดิม การอัปเดตระบบเดิมให้ดียิ่งขึ้น
- ลดความเสี่ยงทางธุรกิจ เทคโนโลยี ต้นทุนในการพัฒนา และเวลา โดยคำนึงถึงโอกาสที่จะเกิดขึ้น และผลกระทบที่จะได้รับหากเกิดขึ้น
- วางแผนการพัฒนาทั้งระยะสั้นและระยะยาว แต่ไม่มีอำนาจในการกำหนดว่าทีมพัฒนาต้องพัฒนาอย่างไร

Developer

- ประกอบไปด้วยตำแหน่งย่อย ดังนี้ Business Analyst, UX Designer, UI Designer, Developer, Quality Assurance, Operation โดยหนึ่งคนอาจมีบทบาทได้มากกว่า 1 ตำแหน่ง
- พัฒนาผลิตภัณฑ์โดยยึดตาม User Story โดยงานจะเสร็จเมื่อ User Story นั้น ผ่าน Acceptance Criteria
- พัฒนา Unit Test และ Automate Test สำหรับ Acceptance Criteria
- พัฒนาระบบ Continuous Integration และ Continuous Delivery เพื่อให้สามารถรวบรวมผลิตภัณฑ์จากทีมต่างๆและส่งมอบให้ผู้ใช้งานได้ง่ายและรวดเร็ว
- พัฒนาทักษะเทคนิคคอล โดยทำวิจัยหรือ Proof of Concept (POC) ก่อนลงมือพัฒนาจริง และ พัฒนา Internal Tools เพื่อช่วยในการทำงานให้ง่ายและเร็วมากขึ้น
- รู้ขีดจำกัดของตัวเองและทีม เช่น Story Point ที่สามารถรับได้ในแต่ละ Sprint

หลักการทำงานแบบ Agile ประกอบด้วย

- มีการทำงานแบบ Cross-functional team คือการนำคนที่มาจากหลายสายงานที่มักมีความต่างกัน มาทำงานร่วมกันอยู่ในทีมเดียวกัน สิ่งนี้จะส่งผลให้ทีมสามารถทำความเข้าใจกับรายละเอียดของงานได้ง่ายขึ้นแล้วยังส่งผลถึงเรื่องการประสานงานกับส่วนงานต่างๆ ที่มีความคล่องตัวมากขึ้น
- ทีมมีอำนาจในการในการตัดสินใจและกำหนดทิศทางของโครงการมากขึ้น ส่วนใหญ่คนที่ได้รับมอบหมายให้ปฏิบัติงานใน Agile squad จะได้รับอำนาจในการตัดสินใจที่มากพอเพื่อไม่ให้โครงการต้องผ่านกระบวนการการขออนุมัติขององค์กรที่มักจะใช้เวลานาน หมายความว่า Product Owner จะต้องมีความตัดสินใจได้ด้วยตัวเอง เพื่อให้เกิดความสะดวกรวดเร็วที่สุด
- ใช้บุคลากรที่ทำงานเพื่อโครงการนี้โดยเฉพาะ มีการแต่งตั้งคนที่รับผิดชอบงานในแต่ละส่วน เพื่อโฟกัสในขอบเขตงานของโครงการที่ได้รับมอบหมายมา
- แบ่งเฟสงานให้เป็นโครงการเล็ก ๆ กำหนดเป้าหมายที่ใช้ระยะเวลาสั้นๆ และต้องส่งมอบผลงานเป็นโครงการเล็ก ๆ เมื่อประเมินผลแล้วว่าอยู่ในทิศทางที่ดีจึงค่อยต่อยอดทำเพิ่มไปเรื่อย ๆ ซึ่งหากพบข้อผิดพลาดหรือจำเป็นต้องมีการเปลี่ยนแปลงใด ก็จะมีการปรับเปลี่ยนการทำงานให้เหมาะสมในแต่ละรอบไป มักเรียกรูปแบบนี้ว่า Sprint
- ทุกคนสามารถรับรู้สถานะของโครงการได้อย่างชัดเจน ทุกคนจะต้องสื่อสารและรับรู้ปัญหาที่เกิดขึ้นของโครงการ รวมทั้งรายงานความคืบหน้าของโครงการให้ทั้งทีมได้รู้ เพื่อทำให้เกิดความชัดเจนและการวัดผลได้

- เกิดการเรียนรู้อยู่เสมอ เรียนรู้ข้อผิดพลาดและข้อดีได้อย่างรวดเร็ว เนื่องจากการทำงานเป็นรอบเล็ก ๆ ทำให้เกิดการเรียนรู้ข้อผิดพลาดที่พบจากครั้งก่อน ๆ และสามารถหาข้อบกพร่องตลอดจนข้อดีในการทำงานได้อย่างรวดเร็ว

สิ่งที่ได้จากการทำ Agile

- ลดความเข้าใจผิดพลาด หากคนในทีมเข้าใจงานคนละอย่าง อาจทำให้ผลลัพธ์ของงานไม่ตอบสนองต่อเป้าหมายที่ต้องการ จึงควรสื่อสารกันอย่างสม่ำเสมอ เพื่อให้ทุกคนเข้าใจตรงกันอยู่เสมอ
- ลดกระบวนการที่ล่าช้า การติดต่อสื่อสารผ่านคนกลางควรเปลี่ยนให้ Dev ได้คุยกับ PO หรือ User โดยตรง และอัปเดตสิ่งที่คุยมาให้ทุกคนในทีมทราบและเข้าใจตรงกัน
- ลดการทำงานที่ซ้ำซ้อนกันงานบางอย่างอาจมีการทำซ้ำซ้อนกับทีมอื่น งานที่เคยมีคนทำไว้แล้ว ควรสามารถหยิบมาใช้หรือพัฒนาต่อยอดได้ โดยไม่ต้องเริ่มใหม่ทั้งหมด

2.3.2 Jira

Jira เป็นเครื่องมือจัดการโปรเจกต์ที่จะคอยเข้ามาช่วยจัดการ Bug Tracking ในงานของเรารวมถึง Plan, Track, Release และทำ Report ปัจจุบันถูกนำมาใช้ในการจัดการ Tracking Agile Project Management กันอย่างแพร่หลาย

ประโยชน์ของการใช้งาน Jira

นอกเหนือจากประสิทธิภาพที่เพิ่มขึ้นแล้ว ระบบ Task Management ที่ดีก็สามารถช่วยบรรเทาสาเหตุของปัญหาต่าง ๆ ในองค์กรที่เราพูดถึงก่อนหน้านี้ได้เช่นกัน เพราะระบบการทำงานที่ดีจะช่วยส่งเสริม Core Value หรือคุณค่าระดับองค์กรในด้านต่างๆ ต่อไปนี้

1. Clarity สร้างการสื่อสารที่ชัดเจน

การใช้ช่องทางการสื่อสารหลากหลายเช่น โทรศัพท์, อีเมล, LINE ทำให้ข้อมูลกระจาย การมีศูนย์กลางในการสื่อสารผู้ใช้แลกเปลี่ยนข้อมูล ส่งต่องาน และอัปเดตสถานะงานในระบบเดียว ช่วยให้ค้นหาข้อมูลได้ถูกต้องและรวดเร็ว

2. Transparency สร้างความโปร่งใสในการทำงาน

ความเสียหายขององค์กรบางครั้งเกิดจากการปิดบังความผิดพลาดและไม่ยอมเปิดเผยข้อมูลอย่างตรงไปตรงมา หรืออาจมีกล่าวโทษกันทำให้ส่งผลถึงขวัญกำลังใจของคนในทีมโดยทั่วไป การสอบสวนสาเหตุของปัญหาย้อนหลังนั้นยาก แต่ Jira จะบันทึกกิจกรรมของงานไว้อย่างชัดเจน ทำให้สามารถตรวจสอบหาต้นตอความผิดพลาดง่ายขึ้น

3. Accountability สร้างความรู้สึกร่วมส่วนร่วมและความรับผิดชอบ

ในองค์กรขนาดใหญ่ที่มีการทำงานหลายระดับ คนทำงานอาจไม่เข้าใจในเป้าหมายที่มาร่วมกัน การใช้ Jira จะทำให้สมาชิกเห็นภาพรวมของการทำงาน ติดตามงานได้ง่าย ทำให้

แต่ละคนเห็นสถานการณ์ที่ตนต้องรับผิดชอบและผลกระทบต่องานของผู้อื่น กระตุ้นให้ตระหนักถึงความสำคัญในส่วนของตนเอง

4. Measurability สร้างการประเมินประสิทธิภาพที่วัดผลได้

ความสำเร็จหรือความล้มเหลวของงานควรเป็นสิ่งที่วัดผลได้ ซึ่ง Jira สามารถรวบรวมและสรุปข้อมูลที่เกิดขึ้นตามจริงเพื่อใช้ชี้วัดประสิทธิภาพของทีมด้านต่าง ๆ เป็นข้อมูลที่แม่นยำสำหรับหัวหน้าหรือผู้บริหารสำหรับประเมินประสิทธิภาพของทีม

2.3.3 Miro

Miro คือไวท์บอร์ดแพลตฟอร์มไดอะแกรม ที่นำมาใช้เป็นเครื่องมือการทำงานร่วมกัน โดยผู้ใช้งานสามารถเลือกเทมเพลตในการใช้งานได้อย่างหลากหลาย ซึ่งถูกนำมาใช้เพื่อสนับสนุนการทำงานแบบ Work-From-Home หรือการทำงานออนไลน์ที่พนักงานทำงานอยู่คนละที่ โดยที่การใช้แคปโปรแกรมประชุมทางไกลไม่ว่าจะเป็น Zoom, Microsoft Team หรือ Google Meet มาใช้ในการพูดคุยหรือนำเสนองานและไอเดียต่าง ๆ ยังไม่ตอบโจทย์การทำงานได้ทั้งหมด จึงเป็นการนำมาของ Miro ตัวช่วยในการไอเดียลงในแพลตฟอร์มไวท์บอร์ดได้อย่างไม่มีขีดจำกัด สร้างการมีส่วนร่วมและเก็บทุกความคิดไว้ไม่ให้ตกหล่น และที่สำคัญคือการเสียเวลามาคอยจดยางงานการประชุมเพราะทุกอย่างจะถูกบันทึกไว้บนไวท์บอร์ดนั้น และพนักงานทุกคนสามารถแสดงความคิดเห็นให้แต่ละไอเดียได้ ทำให้การจัดการความคิดทำได้ง่ายขึ้นและรวดเร็วขึ้น

2.4 ประเภทและระดับของการทดสอบ

2.4.1 Manual Testing

Manual Testing ก็คือ การทดสอบที่ดำเนินการโดยไม่ได้ใช้เครื่องมืออัตโนมัติ (Automated Tool) หรือสคริปต์ (Script) เป็นการจำลองว่าเราคือผู้ใช้ เพื่อกรอกข้อมูล และทำรายการทีละขั้นตอนตามที่ได้ออกแบบไว้ ถ้าเป็นโปรเจกต์หรืองานใหญ่ๆละก็ จะทำให้ใช้เวลาในการทำงานมากขึ้น เพราะการใช้ Manual Test จะต้องใช้แรงงานบุคลากรที่ทำหน้าที่ Testing เข้ามาทดสอบเองทีละเรื่องทีละขั้นทำให้ใช้เวลาค่อนข้างนาน แต่ทั้งนี้ทั้งนั้นการทำ Manual testing จะทำให้เห็นถึงข้อบกพร่องได้อย่างชัดเจน และสามารถส่งเรื่องต่อเพื่อทำการแก้ไข และนำมา re-Test อีกรอบได้อย่างทันที

ข้อดีของ Manual Testing

- คุ่มค่าในระยะสั้น มีประสิทธิภาพการทำงานสูง คาดการณ์วางแผนได้
- ประหยัดค่าใช้จ่าย หรือลงทุนน้อยกว่า Automated Test
- สามารถหาข้อบกพร่องได้อย่างชัดเจน เนื่องจากใช้แรงงานบุคลากรมาทำการทดสอบในแต่ละกรณี

ข้อเสียของ Manual Testing

- ใช้เวลานานในการ Test ค่อนข้างมาก เพราะเป็นการใช้คนมาคอยทำการทดสอบและต้องทำซ้ำ ๆ ไปมาจนกว่าจะไม่เจอข้อผิดพลาด
- ไม่สามารถเปรียบเทียบหรือทำการทดสอบข้อมูลที่มีจำนวนที่เยอะมากหลายพันข้อมูลได้เท่ากับ Automation Testing

2.4.2 Functional Testing

Functional Test เป็นการทดสอบซอฟต์แวร์ประเภทหนึ่งที่ตรวจสอบความถูกต้องของระบบซอฟต์แวร์โดยเทียบกับข้อกำหนด/ข้อกำหนดการใช้งาน วัตถุประสงค์ของการทดสอบฟังก์ชันคือการทดสอบแต่ละฟังก์ชันของแอปพลิเคชัน โดยการป้อนข้อมูลที่เหมาะสม ตรวจสอบผลลัพธ์กับข้อกำหนดด้านฟังก์ชัน

การทดสอบการทำงานส่วนใหญ่เกี่ยวข้องกับการทดสอบกล่องดำและไม่กังวลเกี่ยวกับ Source Code ของแอปพลิเคชัน การทดสอบนี้จะตรวจสอบส่วนต่อประสานผู้ใช้, API, ฐานข้อมูล, ความปลอดภัย, การสื่อสาร Client/Server และการทำงานอื่น ๆ ของแอปพลิเคชันภายใต้การทดสอบ การทดสอบสามารถทำได้ด้วยตนเองหรือใช้ระบบอัตโนมัติ

วัตถุประสงค์หลักของการทดสอบ คือการตรวจสอบการทำงานของระบบซอฟต์แวร์ ส่วนใหญ่มุ่งเน้นไปที่

- ฟังก์ชัน Mainline: การทดสอบฟังก์ชันหลักของแอปพลิเคชัน
- การใช้งานขั้นพื้นฐาน: เกี่ยวข้องกับการทดสอบการใช้งานขั้นพื้นฐานของระบบ โดยจะตรวจสอบว่าผู้ใช้สามารถนำทางผ่านหน้าจอได้อย่างอิสระโดยไม่มีปัญหาใด ๆ
- การเข้าถึง: ตรวจสอบการเข้าถึงของระบบสำหรับผู้ใช้
- เงื่อนไขข้อผิดพลาด: การใช้เทคนิคการทดสอบเพื่อตรวจสอบเงื่อนไขข้อผิดพลาด ตรวจสอบว่ามีการแสดงข้อความแสดงข้อผิดพลาดที่เหมาะสมหรือไม่

วิธีการทดสอบฟังก์ชัน

- ทำความเข้าใจข้อกำหนดการใช้งาน
- ระบุ Input ทดสอบหรือข้อมูลการทดสอบตามข้อกำหนด
- คำนวณผลลัพธ์ที่คาดหวังด้วยค่า Input ทดสอบที่เลือก
- ดำเนินการกรณีทดสอบ
- เปรียบเทียบผลลัพธ์ที่คาดหวังจริงและที่คำนวณได้

2.4.3 Automation Testing

Automation Testing คือการ Test โดยใช้ Tool สำหรับ Test ทำงานด้วยตัวของมันเอง เช่น การเขียนโค้ดขึ้นมาเพื่อให้คอมพิวเตอร์หรือโปรแกรมอื่น ๆ รันการทดสอบด้วยตนเอง โดยไม่ต้องใช้

แรงงานคนในการทำการทดสอบซึ่งวิธีนี้จะช่วยลดเวลาในการ Manual Test แต่อาจจะต้องมีการเขียนโค้ดทำสคริปต์เพื่อรันการทดสอบแบบอัตโนมัติ Automation Testing จึงไม่เหมาะ กับโปรเจกต์สั้น ๆ เพราะว่าส่วนใหญ่แล้วเวลาที่เรานำไปกับการเตรียมการจะมากกว่าที่เราจะได้รับกลับมา

ข้อดี Automation Testing

- ถ้าเราต้องรันการทดสอบซ้ำๆ Automation Testing จะใช้งานได้ดีกว่าไม่ต้องใช้เวลานาน เพราะใช้โปรแกรมที่เขียนโค้ดไว้รันการทดสอบให้เองอัตโนมัติ
- ในการ Test แต่ละครั้งสามารถสร้าง User จำนวนมากขึ้นมาทำการ Test พร้อมกันได้ในเวลาเดียวกัน
- จะใช้เวลาในการ Test สั้นกว่า Manual Testing เนื่องจากมีระบบอัตโนมัติที่ได้มีการเขียนโค้ดไว้

ข้อเสีย Automation test

- ลงทุนสูงในช่วงเริ่มต้น เพราะจำเป็นต้องจ้างบุคคลที่มีความสามารถในการเขียนโค้ด รู้ภาษาคอมที่หลากหลายมาทำการเขียนโค้ดสคริปต์ต่างๆเพื่อรันกรณีทดสอบ
- ต้องใช้ผู้ที่มีทักษะในการเขียนโปรแกรม Test อย่างดี
- Automate test ไม่สามารถทำได้ทุกงาน บางครั้งก็ต้องใช้ Manual Test

2.4.4 System integration Testing

System integration Testing คือ การทดสอบเพื่อตรวจสอบว่าระบบต่างๆ สามารถทำงานร่วมกันได้อย่างถูกต้อง ตรงตามวัตถุประสงค์ ทั้ง Network Integration และ Product Integration ซึ่งจะรวมไปถึง Infrastructure ของระบบ โดยการนำ Service ทุกตัวมารันที่ Server จริง และตรวจสอบให้มั่นใจว่า Service ต่าง ๆ จะต้องทำงานได้ดี สามารถเชื่อมต่อกับ Network, Database, Service อื่น ๆ รวมถึง Party ต่าง ๆ ที่ทีมพัฒนาขึ้นมาได้อย่างถูกต้อง

2.4.5 User Acceptance Testing

User Acceptance Testing คือการให้ Stakeholders เช่น ทีมพาร์ทเนอร์หรือทีม CS ทำการทดสอบบนเครื่อง UAT ก่อนการนำไปติดตั้งบนเครื่อง Production สำหรับการใช้งานจริง เพื่อตรวจสอบว่าตรงกับความต้องการของลูกค้า และตรงตาม Business โฟลว์จริง ๆ ของธุรกิจในระดับที่ยอมรับได้หรือไม่ ซึ่งได้ร่วมกันกำหนดขึ้นมาระหว่างธุรกิจ กับทีมที่พัฒนาระบบ รวมถึงส่วนงานอื่น ๆ ที่เกี่ยวข้อง โดยจะทดสอบในสภาพแวดล้อม (Environment) ที่ใกล้เคียงกับการใช้งานจริงมากที่สุด

สำหรับการทดสอบในขั้นตอนนี้ แตกต่างจากขั้นตอนอื่นๆ คือ ผู้ใช้งานระบบจริงจะต้องเข้ามามีส่วนร่วมในกระบวนการทดสอบซอฟต์แวร์โดยเริ่มตั้งแต่ กำหนดกรณีทดสอบ (Test Case / Scenario) จนถึงการประเมินและสรุปผลการทดสอบ และตัดสินใจว่าระบบดังกล่าวจะสามารถนำไปใช้งานจริง

ได้หรือไม่ ถ้าได้ก็จะมีการลงนามอนุมัติจากเจ้าของโครงการเป็นลายลักษณ์อักษร เพื่ออนุญาตให้นำระบบไปใช้งานจริง แต่ถ้าใช้งานไม่ได้หรือไม่ตรงตามที่ตกลง ระบบจะถูกนำไปปรับปรุงแก้ไข แล้วจะกลับเข้าสู่กระบวนการ UAT ใหม่อีกครั้งจนกว่าจะผ่าน

บทที่ 3

วิธีการดำเนินงานวิจัย

สำหรับฟีเจอร์ควบคุมการใช้งานบัตรเครดิตหรือ “Spending control” เป็นฟีเจอร์ที่เริ่มพัฒนาขึ้นมาใหม่ ทำให้มีการสร้าง API ใหม่และมีการพัฒนาส่วนเชื่อมประสานกับผู้ใช้ใหม่ที่ยังไม่เคยมีการทดสอบมาก่อน ทางผู้จัดทำได้มีหน้าที่ในการทดสอบ API และส่วนเชื่อมประสานกับผู้ใช้ของฟีเจอร์ Spending Control ด้วยการทดสอบแบบ Functional Testing ในระดับ System Integration Test และระดับ User Acceptance Test ตั้งแต่ต้นจนจบโฟลว์โดยมีระยะเวลาและวิธีการดำเนินการดังนี้

1 Training

ในช่วงระยะเวลานี้จะแบ่งออกเป็น 2 ช่วง คือ

- 1.1 ศึกษาเครื่องมือต่าง ๆ ที่ใช้สำหรับการทดสอบ โดยในช่วงเวลานี้มีการศึกษาเครื่องมือที่ใช้สำหรับทดสอบ และเครื่องมือที่ใช้ติดต่อสื่อสารกันภายในทีม รวมไปถึงการตั้งค่าอุปกรณ์ให้สามารถใช้งานได้กับเครื่องมือที่ต้องการใช้งาน
- 1.2 ศึกษาโฟลว์ของงาน โดยในช่วงเวลานี้มีการศึกษาโฟลว์ของพีเจอร์ “Spending Control” ล่วงหน้า โดยมีการดูแลจากผู้ดูแลภายในทีม

2 ทดสอบพีเจอร์ “Spending Control”

สำหรับพีเจอร์ Spending Control นั้น ใช้ระยะเวลาพัฒนาทั้งสิ้น 3 เดือน หรือ 6 Sprint โดยในแต่ละ Sprint จะมีการพูดคุยถึงงานที่ต้องทำ และต้องมีการประเมินระยะเวลาของงานซึ่งมีการทดสอบควบคู่ไปพร้อมกับการพัฒนา โดยทางผู้จัดทำได้แบ่งระยะนี้ออกเป็น 2 ช่วง คือ

- 2.1 ทดสอบด้วย Manual Testing ในการทดสอบระยะนี้ จะเป็นการทดสอบควบคู่ไปพร้อมกับการพัฒนา โดยในช่วงเวลานี้ นั้นจะมีการทดสอบ API หรือ Client บนแต่ละแพลตฟอร์มตามงานที่วางไว้ในแต่ละ Sprint ซึ่งการทดสอบในระยะนี้นั้นมีขั้นตอนดังนี้
 - 2.1.1 ศึกษารายละเอียดของงาน เป็นการศึกษาว่าในแต่ละงานนั้นมีความต้องการอย่างไร และเราควรจะทำแบบ Test Case อย่างไรให้ครอบคลุมและเหมาะสม
 - 2.1.2 เขียน Test Case หลังจากที่ได้ศึกษารายละเอียดของงานแล้ว เราจึงมาสร้างกรณีทดสอบหรือ Test Case หลังจากเขียนเสร็จแล้วจะเข้าสู่กระบวนการพิจารณาจากผู้ดูแล
 - 2.1.3 ทดสอบในระดับ System Integration Test หรือ SIT ซึ่งขั้นตอนนี้จะเริ่มหลังจากที่ผู้พัฒนาทำงานแล้วเสร็จ ซึ่งจะแบ่งเป็นการทดสอบบน Client และการทดสอบ API โดยการทดสอบ Client นั้น จะเป็นการทดสอบบนแพลตฟอร์ม iOS และ Android โดยเมื่อผู้พัฒนาดำเนินการแล้วเสร็จ เราจึงติดตั้งแอปพลิเคชันลงบนแพลตฟอร์มที่กำหนด เพื่อเป็นการทดสอบงานที่เราเพิ่งสร้างหรือมีการอัปเดตขึ้นมาใหม่นั้นเมื่อรวมเข้ากับเซอร์วิสอื่น ๆ ในระบบแล้วยังทำงานถูกต้องตรงตามความต้องการหรือไม่ หรือเมื่อทำการติดตั้งลงบนสมาร์ตโฟนแล้ว ยังทำงานได้ถูกต้องตรงตามความต้องการหรือไม่ และในส่วนของการทดสอบ API เมื่อผู้พัฒนาได้ดำเนินการแล้วเสร็จ จะนำ Endpoint ของเส้นดังกล่าวมาทำการทดสอบด้วย Postman โดยในการทดสอบระดับนี้นั้น จะเป็นการโพส์การทดสอบไปที่ API แต่ละเส้นที่เพิ่มเข้ามาใหม่ และในการทดสอบระดับนี้ยังต้องใช้เครือข่ายภายในองค์กรในการทดสอบ

2.1.4 ทดสอบในระดับ User Acceptance Test หรือ UAT การทดสอบระดับนี้จะเริ่มขึ้นหลังจากทำการทดสอบบน SIT เสร็จสิ้น ซึ่งในการทดสอบ Client บน UAT นั้น ก็จะเป็น environment จริง ส่วนการทดสอบ API จะเป็นการทดสอบการทำงานแบบทั้งโฟลว์ หรือ End-to-end Testing

2.2 ทดสอบด้วย Automation Testing ในระยะเวลานี้ จะทำการสร้างการทดสอบ API ทั้งหมดที่ได้ทำการสร้างขึ้นใหม่ในพีเจอรนี้ โดยใช้ Postman ในการสร้าง Test Script และใช้ Newman ซึ่งเป็นเครื่องมือของ Postman ในการทดสอบตัว Test Script ที่สร้างขึ้น โดยตัวชุดทดสอบนี้ จะถูกทดสอบขึ้นอัตโนมัติเมื่อมีการ Deploy

3 ทดสอบพีเจอร “Pre-customer”

สำหรับพีเจอร Pre-customer นั้น เป็นพีเจอรใหม่ที่เริ่มพัฒนาหลังจากพัฒนาพีเจอร Spending control เสร็จสิ้น โดยมีระยะเวลาพัฒนาทั้งสิ้น 3 เดือน หรือ 6 Sprint โดยระยะแรกของการพัฒนานั้นจะมีการทดสอบแบบ Manual Test ควบคู่ไปกับการพัฒนา โดยขั้นตอนและวิธีการดำเนินงานสำหรับพีเจอรนี้ก็เป็นเช่นเดียวกันกับการทดสอบพีเจอร Spending Control

3.2 Use Case ของพีเจอร Spending Control

พีเจอร Spending Control นั้น เป็นพีเจอรสำหรับควบคุมการใช้งานบัตรเครดิตในเครื่องกรุงศรี โดยผู้ใช้งานสามารถเลือกควบคุมการใช้งานได้ 4 ประเภทได้แก่ การควบคุมการรูดบัตรเครดิตที่หน้าร้าน หรือเรียกว่า CARD การควบคุมวงเงินการใช้จ่ายทางออนไลน์และการตัดเงินรายเดือน หรือเรียกว่า ONLINE การควบคุมการกดเงินจากตู้ ATM หรือเรียกว่า ATM การควบคุมการใช้จ่ายผ่านคิวอาร์โค้ด หรือเรียกว่า QR โดยพีเจอร Spending Control มีแนวทางดังนี้



รูปที่ 3.1 Use Case Diagram

Use Case Description

ตารางที่ 3.2 ตารางคำอธิบาย Use Case Register Application

หัวข้อ	คำอธิบาย
Use Case ID	UC001
Use Case name	Register Application
Actor	User

Pre-Conditions	เปิด UCHOOSE แอปพลิเคชัน และยอมรับ Policy
Post-Conditions	สามารถเข้าใช้งานแอปพลิเคชันได้ในสถานะสมาชิก
Flow	1 กดปุ่มลงทะเบียน 2 กรอกข้อมูล 3 เข้าใช้งาน
Extension Pionts	

ตารางที่ 3.3 ตารางคำอธิบาย Use Case View quickView Screen

หัวข้อ	คำอธิบาย
Use Case ID	UC002
Use Case name	View quickView Screen
Actor	User
Pre-Conditions	ลงทะเบียนเข้าใช้งานแอปพลิเคชัน UCHOOSE และมีบัตรที่สามารถใช้งานพีเจอร์ Spending Control ได้ โดยบัตรที่ไม่สามารถใช้งานได้จะเป็นบัตรที่ถูกปิดการควบคุมในประเภทการรูดบัตรเครดิตที่หน้าร้าน (CARD) และการควบคุมวงเงินการใช้จ่ายทางออนไลน์และการตัดเงินรายเดือน (ONLINE)
Post-Conditions	ผู้ใช้งานสามารถดูรายละเอียดเบื้องต้นของรายการใช้จ่ายประเภท CARD และ ONLINE
Flow	1 กดปุ่ม “บัญชี” 2 กดปุ่ม “ข้อมูลการจำกัดวงเงิน” ที่ด้านล่างขวา
Extension Pionts	

ตารางที่ 3.4 ตารางคำอธิบาย Use Case Receive Value

หัวข้อ	คำอธิบาย
Use Case ID	UC003
Use Case name	Receive Value

Actor	User
Pre-Conditions	รับค่าวงเงินใช้จ่าย/วัน และรายการใช้จ่าย/วัน จากการควบคุมการรูดบัตรเครดิตที่หน้าร้าน (CARD) การควบคุมวงเงินการใช้จ่ายทางออนไลน์และการตัดเงินรายเดือน (ONLINE)
Post-Conditions	
Flow	เมื่อมีการเรียกใช้งานฟังก์ชัน quickView หรือ ข้อมูลการจำกัดวงเงิน จะทำการแสดงค่าวงเงินใช้จ่าย/วัน และรายการใช้จ่าย/วัน จากการควบคุมการรูดบัตรเครดิตที่หน้าร้าน (CARD) หรือ การควบคุมวงเงินการใช้จ่ายทางออนไลน์และการตัดเงินรายเดือน (ONLINE) โดยหากทั้ง 2 ค่าถูกปิดการควบคุมไว้จะไม่สามารถใช้งาน quickView ได้
Extension Pionts	Receive CARD value , Recive ONLINE value

ตารางที่ 3.5 ตารางคำอธิบาย Use Case Verify Eligible Card

หัวข้อ	คำอธิบาย
Use Case ID	UC004
Use Case name	Verify Eligible Card
Actor	
Pre-Conditions	ผู้ใช้งานจะต้องทำการกดปุ่ม “บัญชี” ที่ Navigation bar ด้านล่าง
Post-Conditions	หลังจากกดปุ่มบัญชี มีการยิง API เส้น V1/account เพื่อใช้ตรวจสอบบัตรของผู้ใช้งาน
Flow	
Extension Pionts	

ตารางที่ 3.6 ตารางคำอธิบาย Use Case Edit Card Control

หัวข้อ	คำอธิบาย
Use Case ID	UC005
Use Case name	Edit Card Control
Actor	User
Pre-Conditions	เข้าใช้งานแอปพลิเคชัน UCHOOSE และมีบัตรที่ผ่านการ Verify แล้ว
Post-Conditions	สามารถควบคุมการใช้งานบัตรเครดิตได้ โดยมีการควบคุม 4 ประเภท คือ การควบคุมการรูดบัตรเครดิตที่หน้าร้าน (CARD) การควบคุมวงเงินการใช้จ่ายทางออนไลน์และการตัดเงินรายเดือน (ONLINE) การควบคุมการกดเงินจากตู้ ATM (ATM) การควบคุมการใช้จ่ายผ่านคิวอาร์โค้ด (QR)
Flow	1 กดปุ่ม “บัญชี” 2 กด panel กลางหน้าจอ และเลือกเมนู “การจำกัดวงเงินของคุณ” 3 เลือกประเภทการจำกัดวงเงินที่ต้องการ
Extension Pionts	Edit CARD, Edit ONLINE, Edit ATM, Edit QR
Trigger	หลังจาก 4 พุ่มของทุกวัน ผู้ใช้งานจะไม่สามารถใช้งานการจำกัดวงเงินได้

3.3 การดำเนินงาน

ในการดำเนินการทดสอบฟีเจอร์ Spending Control ทั้งในแบบ Manual Test และ Automate Test หลังจากเข้าใจโฟลว์แล้ว มีขั้นตอนการดำเนินการดังนี้

3.3.1 สร้าง Test Case

หลังจากทำความเข้าใจกับงานที่ได้รับมอบหมายแล้ว จักออกแบบการทดสอบให้เหมาะสม และครอบคลุมกับงานที่ได้รับ เมื่อออกแบบเสร็จสิ้นแล้วจึงนำ Test Case ดังกล่าวไปจัดเก็บลง Testrail เพื่อใช้สำหรับเก็บผลในต่อไป

3.3.2 ทำการทดสอบ

- การทดสอบแบบ Manual Test ในการทดสอบประเภทนี้ มีทั้งการทดสอบบนสมาร์ตโฟนและทดสอบ API ซึ่งการทดสอบบนสมาร์ตโฟนจะทำการทดสอบควบคู่ไปพร้อมกันทั้ง 2 แพลตฟอร์ม โดยการทดสอบทั้งบนสมาร์ตโฟนและ API จะดำเนินการตามขั้นตอนที่ได้ออกแบบไว้

- การทดสอบแบบ Automate Test ในการทดสอบประเภทนี้ จะเป็นการทดสอบ API โดยจะใช้ Postman ในการเขียน Script สำหรับการทดสอบ ซึ่งต้องทำการสร้าง Collection สำหรับใช้ในการทดสอบ ซึ่งตัว Collection นี้ จะรวบรวม API ที่เกี่ยวข้อง และจัดวางตามลำดับการทำงาน และมีการสร้าง Environment ใช้สำหรับเก็บค่าตัวแปรเพื่อความสะดวกในการใช้งาน และเรียกใช้ในเส้นถัดไป

3.3.3 เก็บผล

หลังจากทำการทดสอบเสร็จสิ้น หากมีปัญหาระหว่างทาง จะทำการปรึกษากันภายในทีมถึงแนวทางแก้ปัญหา ซึ่งถ้าหากต้องเลื่อนงานขึ้นดังกล่าวไป Sprint หน้า จะต้องทำการติด Status Block ใน Test Case ที่เกี่ยวข้องกับงานชิ้นนั้นใน Testrail หากงานชิ้นดังกล่าวผ่านตามความต้องการของระบบที่วางไว้แล้ว จะต้องทำการเก็บรายละเอียดของผลการทดสอบลงในเว็บไซต์ภายในของบริษัท และทำการเปลี่ยน Status ดังกล่าวเป็น Pass ใน Testrail พร้อมทั้งแนบรายละเอียดผลการทดสอบลงไปด้วย เพื่อความง่ายในการย้อนกลับมาดูย้อนหลัง

บทที่ 4

ผลการวิจัยและการอภิปรายผล

ในการทดสอบฟีเจอร์ Spending Control บนแอปพลิเคชัน UCHOOSE ได้มีการทดสอบด้วยวิธีการ Manual Testing และ Automation Testing โดยได้มีการนี้ทดสอบและผลลัพธ์สุดท้ายของการทดสอบดังต่อไปนี้

4.1 การทดสอบด้วย Manual Testing

ตารางที่ 4.1 Test Case สำหรับใช้ในการทดสอบในส่วนของ “การจำกัดวงเงิน” หรือ “LIMIT CARD USAGE”

ID	Title	Precondition	Step	Expected Result	Status
	ทดสอบส่วนเชื่อมประสานกับผู้ใช้บนระบบปฏิบัติการ iOS และ Android				
TC01	[EN] displaying Card control section on tab Account template	open UCHOOSE application and register	1 tab "account"	display "LIMIT CARD USAGE" section	Pass
TC02	[TH] displaying Card control section on tab Account template	open UCHOOSE application and register	1 tab "บัญชี"	display "การจำกัดวงเงินของคุณ" section	Pass
TC03	Verify card control information UI when information size more than 1	open UCHOOSE application and register	1 tab "account" 2 tab "CARD LIMIT INFORMATION" section	show card control information UI and enable dropdown	Pass

TC04	Verify card control information UI when information size equal 1	open UCHOOSE application and register	1 tab "account" 2 tab "CARD LIMIT INFORMATION" section	show card control information UI and enable dropdown	Pass
TC05	[EN] Handle empty on Card Control info page	open UCHOOSE application and register	1 tab "account" 2 tab "CARD LIMIT INFORMATION" section	show empty section with message "No Data"	Pass
TC06	[TH] Handle empty on Card Control info page	open UCHOOSE application and register	1 tab "account" 2 tab "ข้อมูลการจำกัดวงเงิน" section	show empty section with message "ไม่มีข้อมูล"	Pass
TC07	[EN] Handle other error on Card Control info page	open UCHOOSE application and register	1 tab "account" 2 tab "CARD LIMIT INFORMATION" section	show empty section with message "Sorry, unsuccessful download" and Retry button	Pass
TC08	[TH] Handle other error on Card Control info page	open UCHOOSE application and register	1 tab "account" 2 tab "ข้อมูลการจำกัดวงเงิน" section	show empty section with message "ขอภัย โหลดข้อมูลไม่สำเร็จ" and ปุ่มลองอีกครั้ง	Pass
	ทดสอบ API				
TC09	Verify response when cardPresentFlag and cardNotPresentFlag = Y	generate token	call card/control/quickView	should call API card/control/detail success 200 with response by format from spec of API	

TC10	Verify response when cardPresentFlag = Y and cardNotPresentFlag = N	generate token	call card/control/quickView	should call API card/control/detail success 200 with response by format from spec of API	
TC11	Verify response when cardPresentFlag = N and cardNotPresentFlag = Y	generate token	call card/control/quickView	should call API card/control/detail success 200 with response by format from spec of API	
TC12	Verify response when cardPresentFlag and cardNotPresentFlag = N	generate token	call card/control/quickView	should call API card/control/detail and status is 204 (no data)	
ทดสอบการเรียกใช้ API และการแสดงผลที่ถูกต้องบนหน้าจอแสดงผล					
TC13	Calling API named: /card/control/quickView and display card and online section	open UCHOOSE application and register	1 tab "account" 2 tab "CARD LIMIT INFORMATION" section	show card control information UI, enable select type and call the API.	
TC14	Calling API named: /card/control/quickView and display only card section	open UCHOOSE application and register	1 tab "account" 2 tab "CARD LIMIT INFORMATION" section	show card control information UI with only card information, disable select type and call the API.	
TC15	Calling API named: /card/control/quickView and display only online section	open UCHOOSE application and register	1 tab "account" 2 tab "CARD LIMIT INFORMATION" section	show card control information UI with only online information, disable select type and call the API.	

ตารางที่ 4.2 Test Case สำหรับใช้ในการทดสอบในส่วนของ “การแก้ไขรายการควบคุมการใช้จ่าย”

ID	Title	Precondition	Step	Expected Result	Status
	ทดสอบส่วนเชื่อมประสานกับผู้ใช้บนระบบปฏิบัติการ iOS และ Android				
TC01	Verify back button in card and online edit page	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on Card and ONLINE section 2 Click back button on the top right	back to SUMMARY page	Pass
TC02	Verify to enable save button in edit Card page when move the scroll bar	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on Card section 2 move scroll bar	save button is enable	Pass
TC03	Verify to enable save button in edit Card page when move the scroll bar and back to the same position	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on Card section 2 move scroll bar and move back to the same position	save button is enable	Pass
TC04	Verify to disable save button in edit Card page	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on Card	save button is disable	Pass

			section 2 Don't move the scroll bar		
TC05	[EN] Verify Remark in the bottom in edit Card page	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on Card section	*In case your spending limit is set at the credit card limit, the spending limit will be varying upon the current credit card limit.	Pass
TC06	[TH] Verify Remark in the bottom in edit Card page	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on Card section	*กรณีมีการจำกัดยอดใช้จ่ายสูงสุดเท่ากับวงเงินเดิมของบัตรเครดิต ยอดใช้จ่ายที่จำกัดดังกล่าวจะถูกปรับเปลี่ยนไปตามวงเงินของท่านในขณะที่มีการใช้บัตร	Pass
TC07	Verify to enable save button in edit ONLINE page when move the scroll bar	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section 2 move scroll bar	save button is enable	Pass
TC08	Verify to enable save button in edit ONLINE page when move the scroll bar and back to the same position	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section 2 move scroll bar and back to the same position	save button is enable	Pass

TC09	Verify to disable save button in edit ONLINE page	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section 6 Don't move the scroll bar	save button is disable	Pass
TC10	[EN] Verify Remark in the bottom in edit ONLINE page	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section	*In case your spending limit is set at the credit card limit, the spending limit will be varying upon the current credit card limit. *You are allowed to set your spending limit for the recurring transactions charged, monthly or yearly, by overseas merchants only. *No. of transactions and amounts, related to Hotel Reservation, Car Rental or Air Ticket that merchants have not yet submitted to the company on transaction date, will be counted under your Spending Limit Control on the merchant submission date.	Pass

TC11	[TH] Verify Remarkin the bottom in edit Oline page	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section	*กรณีมีการจำกัดยอดใช้จ่ายสูงสุดเท่ากับวงเงินเดิมของบัตรเครดิต ยอดใช้จ่ายที่จำกัดดังกล่าวจะถูกปรับเปลี่ยนไปตามวงเงินของท่านในขณะที่มีการใช้บัตร\ก*ท่านสามารถจำกัดยอดใช้จ่ายสำหรับรายการบัตรเครดิตเพื่อชำระค่าสินค้าเป็นรายเดือนหรือรายปี (recurring) ได้เฉพาะรายการจากต่างประเทศเท่านั้น\ก*จำนวนรายการและจำนวนเงินสำหรับการจองโรงแรม, เช่ารถ หรือการซื้อตั๋วเครื่องบิน ที่ทางร้านค้ายังไม่ส่งยอดเข้ามาเรียกเก็บ ณ วันที่ท่านทำรายการ จะถูกนับรวมอยู่ภายใต้การควบคุมรายการใช้จ่ายของท่าน ณ วันที่ทางร้านค้าส่งยอดเข้ามาเรียกเก็บ	Pass
TC12	verify Toggle action in edit ATM page	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ATM section 6 Tab the toggle	- In case tab the on ATM section. The response form card/control/detail should be success 200 with response by format from spec of API	Pass
TC13	verify Toggle action in edit QR pay page	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on QR section 6 Tab the toggle	- In case tab the on QR section. The response form card/control/detail	Pass

				should be success 200 with response by format from spec of API	
	ทดสอบการเรียกใช้ API และการแสดงผลที่ถูกต้องบนหน้าจอแสดงผล				
TC14	verify to update data when edit ATM on limit card usage function	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ATM section 6 Tab the toggle	Should call API : card/control/update success	Pass
TC15	verify to update data when edit QR pay on limit card usage function	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on QR section 6 Tab the toggle	Should call API : card/control/update success	Pass
TC16	Verify the data should be updated after edit data in CARD section	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section 2 Edit data 3 Tab SAVE button	- call API: card/control/update success after tab SAVE button - Navigate to SUMMARY page and call API : card/control/detail - The data should be updated	Pass
TC17	Verify the data should be updated after edit data in ONLINE section	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section	- call API: card/control/update success after tab SAVE button - Navigate to SUMMARY page and call	Pass

			2 Edit data 3 Tab SAVE button	API : card/control/detail - The data should be updated	
TC18	Verify the data should be updated after edit data in ATM section	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ATM section 2 Tab the toggle	- call API: card/control/update success after tab SAVE button - Navigate to SUMMARY page and call API : card/control/detail - The data should be updated	Pass
TC19	Verify the data should be updated after edit data in QR pay section	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on QR section 6 Tab the toggle	- call API: card/control/update success after tab SAVE button - Navigate to SUMMARY page and call API : card/control/detail - The data should be updated	Pass
	ทดสอบการเก็บ Log				
TC20	add transection log when edit CARD in success case	- Generate token	1 call card/control/update 2 Check all data in database	- get response with status 200 and code "0000" - The response should add to database by format from spec of API	Pass
TC21	add transection log when edit ONLINE in success case	- Generate token	1 call card/control/update 2 Check all data in database	- get response with status 200 and code "0000"	Pass

				- The response should add to database by format from spec of API	
TC22	add transection log when edit ATM in success case	- Generate token	1 call card/control/update 2 Check all data in database	- get response with status 200 and code "0000" - The response should add to database by format from spec of API	Pass
TC23	add transection log when edit QR pay in success case	- Generate token	1 call card/control/update 2 Check all data in database	- get response with status 200 and code "0000" - The response should add to database by format from spec of API	Pass
	add transection log when edit CARD in over credit limit case	- Generate token	1 call card/control/update 2 Check all data in database	- get response code "2001" - The response should add to database by format from spec of API	
TC24	add transection log when edit ONLINE in over credit limit case	- Generate token	1 call card/control/update 2 Check all data in database	- get response code "2001" - The response should add to database by format from spec of API	Pass

ตารางที่ 4.3 Test Case สำหรับใช้ในการทดสอบในส่วนของ “การบันทึกรายการควบคุมการใช้จ่าย”

ID	Title	Precondition	Step	Expected Result	Status
----	-------	--------------	------	-----------------	--------

	ทดสอบส่วนเชื่อมประสานกับผู้ใช้บนระบบปฏิบัติการ iOS และ Android				
TC01	[EN]In CARD section. Check errors when editing data during out of service time.	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on CARD section 2 Tab OK	<ul style="list-style-type: none"> - Calling API and getting a response with status code: 503 (Out of Service Hour) - In mobile show alert pop up with message "THE SPENDING LIMIT CONTROL SERVICE IS AVAILABLE\nFROM 01:00 TO 22:00" and OK button - Navigate to summary page after tab OK button 	Pass
	[EN] In ONLINE section. Check for errors when editing data during out of service time.	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section 2 Tab OK	<ul style="list-style-type: none"> - Calling API and getting a response with status code: 503 (Out of Service Hour) - In mobile show alert pop up with message "THE SPENDING LIMIT CONTROL SERVICE IS AVAILABLE\nFROM 01:00 TO 22:00" and OK button - Navigate to summary page after tab OK button 	

TC02	[EN] In ATM section. Check for errors when editing data during out of service time.	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ATM section 2 Tab OK	<ul style="list-style-type: none"> - Calling API and getting a response with status code: 503 (Out of Service Hour) - In mobile show alert pop up with message "THE SPENDING LIMIT CONTROL SERVICE IS AVAILABLE\nFROM 01:00 TO 22:00" and OK button - Navigate to summary page after tab OK button 	Pass
TC03	[EN] In QR section. Check errors when editing data during out of service time.	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on QR section 2 Tab OK	<ul style="list-style-type: none"> - Calling API and getting a response with status code: 503 (Out of Service Hour) - In mobile show alert pop up with message "THE SPENDING LIMIT CONTROL SERVICE IS AVAILABLE\nFROM 01:00 TO 22:00" and OK button - Navigate to summary page after tab OK button 	Pass

TC04	[TH] In CARD section. Check for errors when editing data during out of service time.	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on CARD section 2 Tab OK	<ul style="list-style-type: none"> - Calling API and getting a response with status code: 503 (Out of Service Hour) - In mobile show alert pop up with message "การตั้งรายการจำกัดการใช้วงเงินบัตรเครดิตทำได้ตั้งแต่เวลา 01.00 น. - 22.00 น." and OK button - Navigate to summary page after tab OK button 	Pass
TC05	[TH] In ONLINE section. Check for errors when editing data during out of service time.	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section 2 Tab OK	<ul style="list-style-type: none"> - Calling API and getting a response with status code: 503 (Out of Service Hour) - In mobile show alert pop up with message "การตั้งรายการจำกัดการใช้วงเงินบัตรเครดิตทำได้ตั้งแต่เวลา 01.00 น. - 22.00 น." and OK button - Navigate to summary page after tab OK button 	Pass
TC06	[TH] In ATM section. Check for errors when editing data during out of service time.	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ATM	<ul style="list-style-type: none"> - Calling API and getting a response with status code: 503 (Out of Service Hour) 	Pass

			section 2 Tab OK	- In mobile show alert pop up with message "การตั้งรายการจำกัดการใช้วงเงินบัตรสามารถทำได้ตั้งแต่เวลา 01.00 น. - 22.00 น." and OK button - Navigate to summary page after tab OK button	
TC07	[TH] In QR section. Check for errors when editing data during out of service time.	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on QR section 2 Tab OK	- Calling API and getting a response with status code: 503 (Out of Service Hour) - In mobile show alert pop up with message "การตั้งรายการจำกัดการใช้วงเงินบัตรสามารถทำได้ตั้งแต่เวลา 01.00 น. - 22.00 น." and OK button - Navigate to summary page after tab OK button	Pass
TC08	Edit data on out of service time	open UCHOOSE application and register	1 Select "LIMIT CARD USAGE" icon 2 Display SUMMARY page	pencil button in summary page should be disable	Pass
TC09	In CARD section. Verify to handle case error with status code: 500 and user click cancel	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on CARD section	- display popup alert "Something went wrong. please try again" with cancel	Pass

			2 Edit data 3 Click "save" button 4 Click "CANCEL" of popup	and try again button - navigating to the summary page.	
TC10	In CARD section. Verify to handle case error with status code 204 and user click try again	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on CARD section 2 Edit data 3 Click "save" button 4 Click "TRY AGAIN" of popup	1. Able to display popup alert "Something went wrong. please try again" 2. Able Calling API: card/control/update successful	Pass
TC11	In ONLINE section. Verify to handle case error with status code 500 and user click cancel	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ONLINE section 2 Edit data 3 Click "save" button 4 Click "CANCEL" of popup	- display popup alert "Something went wrong. please try again" with cancel and try again button - navigating to the summary page.	Pass
TC12	In ONLINE section. Verify to handle case error with status code 204 and user click try again	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on CARD section 2 Edit data	- display popup alert "Something went wrong. please try again" with cancel and try again button - Calling API: card/control/update after tab "TRY AGAIN" button	Pass

			3 Click "save" button 4 Click "TRY AGAIN" of popup		
TC13	In ATM section. Verify to handle case error with status code 500 and user click cancel	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ATM section 2 Tab Toggle 3 Click "CANCEL" of popup	- display popup alert "Something went wrong. please try again" with cancel and try again button - navigating to the summary page.	Pass
TC14	In ATM section. Verify to handle case error with status code 204 and user click try again	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on ATM section 2 Tab Toggle 3 Click "TRY AGAIN" of popup	- display popup alert "Something went wrong. please try again" with cancel and try again button - Calling API: card/control/update after tab "TRY AGAIN" button	Pass
TC15	In QR pay section. Verify to handle case error with status code 500 and user click cancel	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on QR section 2 Tab Toggle 3 Click "CANCEL" of popup	- display popup alert "Something went wrong. please try again" with cancel and try again button - navigating to the summary page.	Pass

TC16	In QR pay section. Verify to handle case error with status code 204 and user click try again	open UCHOOSE application and register	1 On SUMMARY page, Click pencil button on QR section 2 Tab Toggle 3 Click "TRY AGAIN" of popup	- display popup alert "Something went wrong. please try again" with cancel and try again button - Calling API: card/control/update after tab "TRY AGAIN" button	Pass
------	--	---------------------------------------	--	--	------

4.2 การทดสอบด้วย Automation Testing

การทดสอบ API เส้น card/customer/list ในกรณีที่สามารถทำงานได้สำเร็จ และสามารถ
ซัพพอร์ตฟีเจอร์ Spending Control ได้

```
"results": [
{
  "name": "200 - CARD_CONTROL_Activate",
  "time": 116,
  "responseCode": {
    "code": 200,
    "name": "OK"
  },
  "tests": {
    "Verify that the response code should be 200": true,
    "Verify that the response code is success": true,
    "Verify that the response message is success": true,
    "Verify card status is ACTIVATE": true,
    "Verify that the response action contain CARD_CONTROL": true
  },
  "testPassFailCounts": {
    "Verify that the response code should be 200": {
      "pass": 18,
      "fail": 0
    },
    "Verify that the response code is success": {
      "pass": 18,
      "fail": 0
    },
    "Verify that the response message is success": {
      "pass": 18,
      "fail": 0
    },
    "Verify card status is ACTIVATE": {
```

```

        "pass": 18,
        "fail": 0
    },
    "Verify that the response action contain CARD_CONTROL": {
        "pass": 18,
        "fail": 0
    }
}
]

```

การทดสอบ API เส้น card/customer/list ในกรณีที่สามารถทำงานได้สำเร็จ แต่ไม่สามารถ
ซัพพอร์ตฟีเจอร์ Spending Control ได้

```

"results": [
{
    "name": "200 - CARD_CONTROL_is_not_Activate",
    "time": 103,
    "responseCode": {
        "code": 200,
        "name": "OK"
    },
    "tests": {
        "Verify that the response code should be 200": true,
        "Verify that the response code is success": true,
        "Verify that the response message is success": true,
        "Verify card status is NOT ACTIVATE": true,
        "Verify that the response action does NOT contain
CARD_CONTROL": true
    },
    "testPassFailCounts": {
        "Verify that the response code should be 200": {
            "pass": 20,
            "fail": 0
        },
    },
}
]

```

```

    "Verify that the response code is success": {
        "pass": 20,
        "fail": 0
    },
    "Verify that the response message is success": {
        "pass": 20,
        "fail": 0
    },
    "Verify card status is NOT ACTIVATE": {
        "pass": 20,
        "fail": 0
    },
    "Verify that the response action does NOT contain
CARD_CONTROL": {
        "pass": 20,
        "fail": 0
    }
}

```

```

}

```

```

]

```

การทดสอบ API เส้น feature/cardControl/list โดย cardList จะต้องสัมพันธ์กับ cardList
ที่ได้จากเส้น card/customer/list และกรอกรสถานะที่ไม่ต้องการออก

```

"results": [
{
    "name": "card/feature/cardControl/list",
    "time": 192,
    "responseCode": {
        "code": 200,
        "name": "OK"
    },
    "tests": {
        "Verify that the response code should be 200": true,

```

```

    "Verify that the response code is success": true,
    "Verify that the response message is success": true,
    "Verify that the response data not empty": true,
    "Verify number of the cardList was equal from number of cardList
in card/customer/list": true,
    "Verify card Number X-5332": true,
    "Verify card Number X-9885": true,
    "Verify card Number X-5834": true
  },
  "testPassFailCounts": {
    "Verify that the response code should be 200": {
      "pass": 6,
      "fail": 0
    },
    "Verify that the response code is success": {
      "pass": 6,
      "fail": 0
    },
    "Verify that the response message is success": {
      "pass": 6,
      "fail": 0
    },
    "Verify that the response data not empty": {
      "pass": 6,
      "fail": 0
    },
    "Verify number of the cardList was equal from number of cardList
in card/customer/list": {
      "pass": 6,
      "fail": 0
    },
    "Verify card Number X-5332": {
      "pass": 1,

```

```

        "fail": 0
      },
      "Verify card Number X-9885": {
        "pass": 1,
        "fail": 0
      },
      "Verify card Number X-5834": {
        "pass": 1,
        "fail": 0
      }
    }
  }
]

```

การทดสอบ API เส้น card/detail โดยเป็นการตรวจสอบ Tag ที่ต้องการแสดงกับสถานะของ Tag จาก V1/account

```

"results": [
{
  "name": "card/detail in eligible case",
  "time": 1392,
  "responseCode": {
    "code": 200,
    "name": "OK"
  },
  "tests": {
    "tag[N,N,Y,N] Verify that the response cardControl quickView is false":
true,
    "Verify that the response cardControl contain ATM": true,
    "Verify that the response code should be 200": true,
    "Verify that the response code is success": true,

```



```

    "Verify that the response message is success": true
  },
  "testPassFailCounts": {
    "tag[Y,Y,N,Y] Verify that the response cardControl quickView is true": {
      "pass": 1,
      "fail": 0
    },
    "Verify that the response cardControl contain CARD,ONLINE,QR": {
      "pass": 1,
      "fail": 0
    },
    "Verify that the response code should be 200": {
      "pass": 12,
      "fail": 0
    },
    "Verify that the response code is success": {
      "pass": 12,
      "fail": 0
    },
    "Verify that the response message is success": {
      "pass": 12,
      "fail": 0
    },
    "tag[N,Y,N,N] Verify that the response cardControl quickView is true": {
      "pass": 1,
      "fail": 0
    }
  }
}

```

```

},
"Verify that the response cardControl contain ONLINE": {
    "pass": 2,
    "fail": 0
},
"tag[Y,N,N,N] Verify that the response cardControl quickView is true": {
    "pass": 1,
    "fail": 0
},
"Verify that the response cardControl contain CARD": {
    "pass": 1,
    "fail": 0
},
"tag[Y,Y,Y,Y] Verify that the response cardControl quickView is true": {
    "pass": 1,
    "fail": 0
},
"Verify that the response cardControl contain CARD,ONLINE,ATM,QR": {
    "pass": 1,
    "fail": 0
},
"tag[N,N,Y,Y] Verify that the response cardControl quickView is false": {
    "pass": 1,
    "fail": 0
},
"Verify that the response cardControl contain ATM,QR": {

```

```

    "pass": 1,
    "fail": 0
  },
  "tag[N,Y,Y,N] Verify that the response cardControl quickView is true": {
    "pass": 1,
    "fail": 0
  },
  "Verify that the response cardControl contain ONLINE,ATM": {
    "pass": 1,
    "fail": 0
  },
  "tag[Y,Y,N,N] Verify that the response cardControl quickView is true": {
    "pass": 1,
    "fail": 0
  },
  "Verify that the response cardControl contain CARD,ONLINE": {
    "pass": 1,
    "fail": 0
  },
  "tag[N,N,N,Y] Verify that the response cardControl quickView is false": {
    "pass": 1,
    "fail": 0
  },
  "Verify that the response cardControl contain QR": {
    "pass": 2,
    "fail": 0
  }

```

```

    },
    "tag[Y,N,Y,N] Verify that the response cardControl quickView is true": {
        "pass": 1,
        "fail": 0
    },
    "Verify that the response cardControl contain CARD,ATM": {
        "pass": 1,
        "fail": 0
    },
    "tag[N,N,Y,N] Verify that the response cardControl quickView is false": {
        "pass": 1,
        "fail": 0
    },
    "Verify that the response cardControl contain ATM": {
        "pass": 1,
        "fail": 0
    }
}
}
]

```

การทดสอบ API เส้น card/detail ในกรณีที่ tag ของการ์ด เป็น NNNN

```

"results": [
{
    "name": "card/detail in eligible case",
    "time": 323,
    "responseCode": {
        "code": 200,

```

```

        "name": "OK"
    },
    "tests": {
        "Verify tag must be NNNN": true,
        "tag[NNNN] Verify that the response cardControl quickView is
false": true,
        "Verify this card is ineligible": true,
        "Verify that the response code should be 200": true,
        "Verify that the response code is success": true,
        "Verify that the response message is success": true
    },
    "testPassFailCounts": {
        "Verify tag must be NNNN": {
            "pass": 46,
            "fail": 0
        },
        "tag[NNNN] Verify that the response cardControl quickView is
false": {
            "pass": 1,
            "fail": 0
        },
        "Verify this card is ineligible": {
            "pass": 46,
            "fail": 0
        },
        "Verify that the response code should be 200": {
            "pass": 46,
            "fail": 0
        },
        "Verify that the response code is success": {
            "pass": 46,
            "fail": 0
        }
    },

```

```

        "Verify that the response message is success": {
            "pass": 46,
            "fail": 0
        }
    }
}
]

```

การทดสอบ API เส้น card/control/quickView โดยเป็นการตรวจสอบค่าที่แสดงในฟังก์ชัน quickView นั้นมีค่าเท่ากับค่าที่แสดงใน emboss

```

"results": [
{
    "name": "card/control/quickView",
    "time": 167,
    "responseCode": {
        "code": 200,
        "name": "OK"
    },
    "tests": {
        "Verify that the response code should be 200": true,
        "Verify that the response code is success": true,
        "Verify that the response message is success": true,
        "Verify type of information is CARD": true,
        "[CARD] Verify value of dailySpendingLimit : 100000": true,
        "[CARD] Verify value of dailySpending : 0": true,
        "[CARD] Verify value of dailySpendingItemsLimit : 200": true,
        "[CARD] Verify value of dailySpendingItems : 0": true
    },
    "testPassFailCounts": {
        "Verify that the response code should be 200": {
            "pass": 8,
            "fail": 0
        },
    },
}
]

```

```

"Verify that the response code is success": {
    "pass": 8,
    "fail": 0
},
"Verify that the response message is success": {
    "pass": 8,
    "fail": 0
},
"Verify type of information is CARD": {
    "pass": 5,
    "fail": 0
},
"[CARD] Verify value of dailySpendingLimit : 14999": {
    "pass": 1,
    "fail": 0
},
"[CARD] Verify value of dailySpending : 0": {
    "pass": 1,
    "fail": 0
},
"[CARD] Verify value of dailySpendingItemsLimit : 20": {
    "pass": 1,
    "fail": 0
},
"[CARD] Verify value of dailySpendingItems : 0": {
    "pass": 1,
    "fail": 0
},
"Verify type of information is ONLINE": {
    "pass": 6,
    "fail": 0
},
"[ONLINE] Verify value of dailySpendingLimit : 13943": {

```

```
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpending : 0": {
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingItemsLimit : 20": {
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingItems : 0": {
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingLimit : 59000": {
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingItemsLimit : 100": {
        "pass": 1,
        "fail": 0
    },
    "[CARD] Verify value of  dailySpendingLimit : 100000": {
        "pass": 1,
        "fail": 0
    },
    "[CARD] Verify value of  dailySpendingItemsLimit : 200": {
        "pass": 1,
        "fail": 0
    },
    "[CARD] Verify value of  dailySpendingLimit : 25525": {
        "pass": 1,
```



```
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingLimit : 137805": {
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingItemsLimit : 200": {
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingLimit : 200000": {
        "pass": 1,
        "fail": 0
    },
    "[CARD] Verify value of  dailySpendingLimit : 191029": {
        "pass": 1,
        "fail": 0
    },
    "[CARD] Verify value of  dailySpendingItemsLimit : 0": {
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingLimit : 233245": {
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingItemsLimit : 1": {
        "pass": 1,
        "fail": 0
    },
    "[ONLINE] Verify value of  dailySpendingLimit : 100000": {
        "pass": 1,
        "fail": 0
    }
```

```

    }
  }
}

]
```

การทดสอบ API เส้น card/control/quickView ในกรณีที่เรียกใช้สำเร็จ แต่ผู้ใช้งานไม่ได้เปิดใช้งาน

```

"results": [
{
  "name": "card/control/quickView",
  "time": 284,
  "responseCode": {
    "code": 204,
    "name": "No Content"
  },
  "tests": {
    "Verify that the response code should be 204": true
  },
  "testPassFailCounts": {
    "Verify that the response code should be 204": {
      "pass": 50,
      "fail": 0
    }
  }
}
]
]
```

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

สำหรับสหกิจศึกษาครั้งนี้ ผู้จัดทำได้เข้ามาเรียนรู้เกี่ยวกับบทบาทของ Quality Assurance หรือ QA ในวิธีการ Agile โดยผลการทดสอบที่ได้แสดงในผลที่ 4 นั้น เป็นผลลัพธ์สุดท้ายที่ได้จากการทดสอบ โดยข้อบกพร่องที่พบเจอในระหว่างการทำงาน จะถูกรายงานให้นักพัฒนาที่เกี่ยวข้องในทันที และทำการแก้ไข จากนั้นจึงจะทำการทดสอบใหม่อีกครั้งเมื่อทำการแก้ไขเสร็จสิ้น

ในการทดสอบพีเจอาร์ Spending Control ผู้จัดทำได้มีส่วนร่วมในการทดสอบทั้งแบบ Manual Testing และ Automation Testing โดยในแบบ Manual Testing นั้นได้มีการทดสอบการแสดงผลผ่านหน้าจอ การทดสอบ API และการทดสอบการเก็บ log ลงฐานข้อมูล โดยการทดสอบแบบนี้ ผู้จัดทำได้มีส่วนร่วมในการทดสอบอยู่ 3 ส่วน ได้แก่

1. “การจำกัดวงเงิน”

โดยในส่วนนี้มีการทดสอบเกี่ยวกับการเพิ่มส่วนของ “การจำกัดวงเงิน” ในหน้าบัญชี และทดสอบการแสดงผลหลังจากทำการแตะที่คำว่า “การจำกัดวงเงิน” โดยความต้องการของในส่วนนี้คือ หากบัญชีผู้ใช้งานดังกล่าว เปิดใช้งานการควบคุมการใช้งานบัตร ทั้งประเภท CARD และ ONLINE โดยต้องมี Drop-down menu ในการเลือกเพื่อแสดงผลประเภทของการควบคุมที่ต้องการ โดยหากบัญชีดังกล่าวเปิดการใช้งานเพียง 1 ประเภท Drop-down menu จะไม่สามารถใช้งานได้ หรือหากบัญชีดังกล่าว ไม่ได้เปิดใช้งานการควบคุม ต้องแสดงคำว่า no data ซึ่งในส่วนของการ “การจำกัดวงเงิน” ได้มีการทดสอบ API เพื่อให้ได้ผลลัพธ์สอดคล้องกับความต้องการ และสามารถนำไปแมพกับการแสดงผลได้อย่างถูกต้อง และเหมาะสม

2. “การแก้ไขรายการควบคุมการใช้จ่าย”

เป็นหน้าสำหรับแก้ไขรายการการใช้จ่ายในแต่ละประเภท โดยทำการตรวจสอบการแก้ไข และปุ่มต่าง ๆ แต่เนื่องจากเป็นหน้าที่เพิ่งสร้างขึ้นใหม่ จะต้องทำการตรวจสอบการใช้จ่าย เพิ่มเข้าไปด้วย โดยในส่วนนี้มีการทดสอบการเรียกใช้ API และทดสอบว่าในแต่ละรายการที่ทำการแก้ไขและบันทึกข้อมูลได้สำเร็จนั้น ได้ถูกบันทึกลงฐานข้อมูลตรงตามรูปแบบที่วางไว้หรือไม่

3. “การบันทึกรายการควบคุมการใช้จ่าย”

ในส่วนนี้ผู้จัดทำได้รับผิดชอบในส่วนของการ Error Case เช่น การบันทึกรายการการแก้ไขหลังจากเวลาให้บริการ หรือ Internet Error โดยจะทดสอบการแสดงผลว่าตรงตามที่ต้องการหรือไม่

และการทดสอบแบบ Automation testing โดยเป็นการทดสอบว่า API เส้นดังกล่าวสามารถใช้งานได้ปกติ และมีโครงสร้างตรงกับความต้องการ และค่าที่ได้มีความถูกต้อง

5.2 ข้อเสนอแนะ

เนื่องจากบทบาทของผู้จัดทำในโครงการสหกิจศึกษาครั้งนี้คือ QA ซึ่งการตรวจสอบในแต่ละส่วนนั้นต้องมีความรู้และความเข้าใจ และอาจจะต้องใช้งาน Service อื่นร่วมด้วยในการทดสอบ ซึ่งวิธีที่จะให้คนมาที่หลังนั้นมีความรู้ความเข้าใจด้วยผู้จัดทำมีความเห็นว่าการทำ Document เป็นตัวเลือกที่ดีที่จะเก็บรายละเอียดของงานต่าง ๆ ไว้โดยไม่สูญหาย อีกทั้งยังลดเวลาในการในการติดต่อสื่อสาร

เอกสารอ้างอิง

วิจิตรา พรหมจรรย์. เรียนรู้คำสั่ง Sql ไปด้วยกัน Step by step. Blog NSTRU [อินเทอร์เน็ต]. [เข้าถึงเมื่อ 4 พ.ย. 2565]. เข้าถึงได้จาก: https://race.nstru.ac.th/home_ex/blog/topic/show/2940

W3schools. SQL Tutorial [อินเทอร์เน็ต]. [เข้าถึงเมื่อ 4 พ.ย. 2565]. เข้าถึงได้จาก: <https://www.w3schools.com/sql/>

TestRail. Comprehensive Test Case Management for Your Team [อินเทอร์เน็ต]. [เข้าถึงเมื่อ 4 พ.ย. 2565]. เข้าถึงได้จาก: <https://www.gurock.com/testrail/tour/modern-test-management/>

Software Testing Help. Top 20 Best Test Management Tools (New 2023 Rankings) [อินเทอร์เน็ต]. 2565 [เข้าถึงเมื่อ 4 พ.ย. 2565]. เข้าถึงได้จาก: <https://www.softwaretestinghelp.com/15-best-test-management-tools-for-software-testers/>

Asustor. Introducing Portainer [อินเทอร์เน็ต]. 2565 [เข้าถึงเมื่อ 6 พ.ย. 2565]. เข้าถึงได้จาก: https://www.asustor.com/th/online/College_topic?topic=145

Puttapong Khemcharoen, การเขียน Test Script ด้วย Postman [อินเทอร์เน็ต]. 2561 [เข้าถึงเมื่อ 6 พ.ย. 2565]. เข้าถึงได้จาก: <https://medium.com/scale360-engineering/>

ThaiCreate Com Team. ติดตั้ง Oracle SQL Developer จัดการกับ Oracle การสร้าง User / Grant สิทธิ์ [อินเทอร์เน็ต]. 2560 [เข้าถึงเมื่อ 10 พ.ย. 2565]. เข้าถึงได้จาก: <https://www.thaicreate.com/tutorial/oracle-sql-developer.html>

Thanyavuth Akarasomcheep. Agile คืออะไร เริ่มใช้งานอย่างไร [อินเทอร์เน็ต]. 2561 [เข้าถึงเมื่อ 14 พ.ย. 2565]. เข้าถึงได้จาก: <https://medium.com/fastwork-engineering/agile->

Trends and Technology. การทำงานแบบ Agile แนวคิดในการทำงานขององค์กรยุคใหม่ [อินเทอร์เน็ต]. 2562 [เข้าถึงเมื่อ 19 พ.ย. 2565] เข้าถึงได้จาก: <https://www.tnt.co.th/news/162-agile-framework-working-principle-for-modern-company>

Data Express. รู้จักและเริ่มใช้งาน Jira Project Management Software (1/4) [อินเทอร์เน็ต]. [เข้าถึงเมื่อ 19 พ.ย. 2565]. เข้าถึงได้จาก: <https://dataexpress.co.th>

Sakul Montha. Jira คืออะไร แล้ว Epic, Story, Task และ Sub-Task ต่างกันอย่างไร [อินเทอร์เน็ต]. 2562 [เข้าถึงเมื่อ 21 พ.ย. 2565]. เข้าถึงได้จาก: <https://iamgique.medium.com/Jira-%>

Leading the Future. Miro : the online collaborative white platform [อินเทอร์เน็ต]. 2564 [เข้าถึงเมื่อ 22 พ.ย. 2565]. เข้าถึงได้จาก: <https://www.1-to-all.com/blog/miro-the-online-collaborativenbspwhiteboard-platform>

MarcusCode. แนะนำภาษา JaVaScript [อินเทอร์เน็ต]. 2563 [เข้าถึงเมื่อ 22 พ.ย. 2565]. เข้าถึงได้จาก: <http://marcuscode.com/lang/javascript/introducing-to-javascript>

Mindphp. ข้อดี / ข้อเสียของ Manual Testing Automated Testing [อินเทอร์เน็ต]. 2562.[เข้าถึงเมื่อ 24 พ.ย. 2565]. เข้าถึงได้จาก: <https://www.mindphp.com/forums/viewtopic.php?f=85&t=57739>

Thomas Hamilton. What is Functional Testing? Types & Examples [อินเทอร์เน็ต]. 2565 [เข้าถึงเมื่อ 24 พ.ย. 2565]. เข้าถึงได้จาก: <https://www.guru99.com/functional-testing.html>

GeniusSoft. ทำความรู้จักกับ SOFTWARE TESTER/QA [อินเทอร์เน็ต]. 2565. [เข้าถึงเมื่อ 25 พ.ย. 2565]. เข้าถึงได้จาก: <https://www.geniussoft.co.th/software-tester-qa-ep1>

Trends and Technology. Software Testing Process ขั้นตอนการทดสอบซอฟต์แวร์ที่ developer ไม่ควรพลาด [อินเทอร์เน็ต]. 2562 [เข้าถึงเมื่อ 27 พ.ย. 2565] เข้าถึงได้จาก: <https://www.tnt.co.th/news/160-would-you-help-software-development-project-to->

Thomas Hamilton. What is System Integration Testing (SIT) with Example [อินเทอร์เน็ต]. 2565 [เข้าถึงเมื่อ 28 พ.ย. 2565]. เข้าถึงได้จาก: <https://www.guru99.com/system-integration-testing.html>

ภาคผนวก



งานทะเบียนคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
คำรับรองเล่มโครงการพิเศษ/ปัญหาพิเศษ/สหกิจศึกษา

วันที่ 10 เดือน กุมภาพันธ์ พ.ศ. 2566

ข้าพเจ้า นางสาว เกษราภรณ์ โคเพื่อง รหัสประจำตัว 62050130 นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ ขอรับรองว่าสหกิจศึกษา เรื่อง ชื่อภาษาไทย การทดสอบฟีเจอร์สำหรับควบคุมการใช้งานบัตรเครดิตบนแอปพลิเคชันยูซูส ชื่อภาษาอังกฤษ UCHOOSE APPLICATION FEATURE TESTING FOR SPENDING CONTROL ปีการศึกษา 2565 เป็นผลงานวิจัยที่ได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อนเรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่ม สหกิจศึกษาฉบับสมบูรณ์แล้ว โปรแกรมอักขราวิสุทธิ์ 0.27%

ลงชื่อ.....เกษราภรณ์ โคเพื่อง.....

(นางสาวเกษราภรณ์ โคเพื่อง)

นักศึกษา

ข้าพเจ้า ดร. อัครเดช อุดมชัยพร อาจารย์ที่ปรึกษาสหกิจศึกษา ได้ตรวจสอบโครงการสหกิจศึกษาของนักศึกษาข้างต้น แล้ว ขอรับรองว่าเป็นผลงานวิจัยของนักศึกษาจริงและมีเนื้อหาสมบูรณ์ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ.....อัครเดช.....

อาจารย์ที่ปรึกษา

Plagiarism Checking Report

Created on Feb 8, 2023 at 23:01 PM

Submission Information

ID	SUBMISSION DATE	SUBMITTED BY	ORGANIZATION	FILENAME	STATUS	SIMILARITY INDEX
2959160	Feb 8, 2023 at 23:01 PM	62050130@kmitl.ac.th	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	62050130_การทดสอบพีเจอาร์สำหรับควบคุมการใช้งานบัตรเครดิตบน แอปพลิเคชันยูซูส .pdf	Completed	0.27 %

Match Overview

NO.	TITLE	AUTHOR(S)	SOURCE	SIMILARITY INDEX
1	no metadata	no metadata	จุฬาลงกรณ์มหาวิทยาลัย	0.27 %

Match Details

TEXT FROM SUBMITTED DOCUMENT	TEXT FROM SOURCE DOCUMENT(S)
งสารบัญบทคัดย k อกบทคัดย k ภาษาอังกฤษชกิตติกรรมประกาศสารบัญง สารบัญตารางงสารบัญรูปขบพที่ 1 บทนำ 1 11 ความเป j นมาและความสำคัญ ของปัญหา 1 12 วัตถุประสงค์ของงานวิจัย 1 13 ขอบเขตของงานวิจัย 2 14 ประโยชน์ที่คาว k ะได้ 8 รับ 3 บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4 21 ภาษาโปรแกรมมิ่งที่ไ 8 ในการทดสอบซอฟต์แวร์ 4 211 JavaScript Language 4 212ฐ\ n บทที่ 1 บทนำ\ n\ n ความเป นมาและความสำคัญของป ญหา 1 \\ n คำถามการวิจัย 5 \ n วัตถุประสงค์ ของการวิจัย 5 \ n ระเบียบวิธีวิจัย 5 \\ n ขอบเขตของการวิจัย 5 \ n ข อดกลงเบื้องต นสำหรับการวิจัย 7 \ n คำจ่า กัดความที่ไ ใในการวิจัย 7 \ n ประโยชน ที่คาว ะได้ รับ 8 \ n \ n บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง \ n ตอนที่ 1 วิชาสังคตินิยม 10 \ n \ n 1.1 ความหมายของวิชาสังคตินิยม 10 \ n 1.2 จุดประสงค์ ของการเรียนการ สอนวิชาสังคตินิยม 11 \ n 1.3 สาระการเรียนวิชาสังคตินิยม 13 \ n 1.4 สือที่ ไ ใในการเรียนการสอนวิชาสังคตินิยม 14 \ n 1.5 การวัดและประเมินผลการ เรียนวิชาสังคตินิยม 15 \ n 1.6 วิชาสังคตินิยมในระดับอุดมศึกษาใน ประเทศไทย 16 \ n \ n ตอนที่ 2 สุนทรียภาพทางดนตรี 18 \ n 2.1 ความ หมายของสุนทรียภาพและคุณค ะของสุนทรียประสภารณ 18 \ n 2.2 องค์ ประกอบของสุนทรียประสภารณ 20 \ n 2.3 องค์ ประกอบใน เรื่งของความซาบซัง ในดนตรี 21 \ n \ n \ n \ n บทที่หน ะ \ n \ n \ n 2.4 การพัฒนาสุนทรียภาพทางดนตรี 23 \ n 2.5 อุปสรรคต อดความซาบซัง ในสุนทรียภาพทางดนตรีและแนว \ n \ n ทางแก

คำรับรองเล่มสหกิจศึกษาโดยสถานประกอบการ

วันที่ 20 เดือน มีนาคม พ.ศ. 2565

ข้าพเจ้า นางสาวภาวีนี ไชยสฤติย์ ตำแหน่ง Manager เป็นตัวแทนของสถานประกอบการ บริษัท อยุธาแคปปิตอล เซอร์วิสเชส จำกัด ขอรับรองว่า ทางสถานประกอบการได้ตรวจสอบเล่มสหกิจศึกษา เรื่อง การทดสอบฟีเจอร์สำหรับควบคุมการใช้งานบัตรเครดิตบนแอปพลิเคชันยูซูส (UCHOOSE APPLICATION FEATURE TESTING FOR SPENDING CONTROL) ของนักศึกษาชื่อ นางสาวเกษราภรณ์ โคเฟื่อง ซึ่งเป็นนักศึกษาภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เรียบร้อยแล้ว และไม่มีส่วนหนึ่งส่วนใดในเล่มสหกิจศึกษานี้ที่มีข้อมูลอ่อนไหว และ/หรือ ข้อมูลอันเป็นความลับอันจะก่อให้เกิดความเสียหายต่อสถานประกอบการ รวมทั้งอนุญาตให้สามารถเผยแพร่ต่อสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังได้ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ ภาวีนี ไชยสฤติย์

(นางสาวภาวีนี ไชยสฤติย์)

ตัวแทนสถานประกอบการ

ข้าพเจ้า ดร.อัคเดช อุดมชัยพร อาจารย์ที่ปรึกษาสหกิจศึกษา ได้ตรวจสอบเล่มสหกิจศึกษาแล้วและรับทราบว่าสถานประกอบการดำเนินการตรวจสอบเล่มสหกิจศึกษาแล้ว จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ อัคเดช

(ดร.อัคเดช อุดมชัยพร)

อาจารย์ที่ปรึกษา