

โครงการ ระบบห้องสมุด (Library System)											
รหัสสถานการณ์การทดสอบ : SC01		ผู้เขียนสถานการณ์การทดสอบ : นางสาวสาวลักษณ์ บุญรอด									
ฟังก์ชันที่ทดสอบ : Book() Book(String title, String author), getTitle(), getAuthor(), isBorrowed(), borrow(), returnBook()		วันที่เขียน : 7 ตุลาคม 2568									
เงื่อนไขก่อนทดสอบ : มีการสร้างอ้อมเจกต์ Book สำเร็จแล้ว เช่น new Book("Software testing", "Yogesh Singh")											
เงื่อนไขหลังทดสอบ : สถานะการยืม (borrowed) ของหนังสือเปลี่ยนแปลงถูกต้องตามที่คาดไว้ หากมีการยืมหรือคืนซ้ำ ต้องไม่มีการเกิด Error ที่ไม่คาดคิด											
ขั้นตอนการดำเนินการทดสอบ :											
<ol style="list-style-type: none"> ทดสอบการสร้างหนังสือที่มีชื่อและผู้แต่งปกติเปิดໂປຣເຈກຕີທີ່ມີ package library ทดสอบการสร้างหนังสือที่มี title เป็นค่าว่าง, null, หรือช่องว่าง ทดสอบเมธອດ borrow() ให้เปลี่ยนสถานะเป็น borrowed ทดสอบการยืมซ้ำเพื่อให้เกิด IllegalStateException ทดสอบ returnBook() เพื่อคืนหนังสือ ทดสอบการคืนซ้ำหลายครั้ง (ต้องไม่เกิดข้อผิดพลาด) ตรวจสอบค่าที่ getter (getTitle, getAuthor, isBorrowed) คืนค่าถูกต้อง 											
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ					
TC01	สร้างหนังสือใหม่	new Book("Software Testing", "Yogesh Singh")	สร้างอ้อมเจกต์ Book ได้สำเร็จ ไม่มี Error	ระบบสร้างอ้อมเจกต์ได้สำเร็จ และคงค่า title และ author ถูกต้อง	ผ่าน						
TC02	สร้างหนังสือด้วย title ว่าง	("", "Unknown")	Throw IllegalArgumentException	ระบบแจ้ง Exception ตามที่คาด ไม่มี Error อื่น	ผ่าน						
TC03	สร้างหนังสือด้วย title = null	(null, "Unknown")	Throw IllegalArgumentException	ระบบแจ้ง Exception ตามที่คาด ไม่มี Error อื่น	ผ่าน						
TC04	สร้างหนังสือด้วยช่องว่าง	(" ", "Anonymous")	Throw IllegalArgumentException	ระบบแจ้ง Exception ตามที่คาด ไม่มี Error อื่น	ผ่าน						
TC05	ยืมหนังสือครั้งแรก	book.borrow()	isBorrowed = true	ระบบเปลี่ยนสถานะ borrowed เป็น true ถูกต้อง	ผ่าน						
TC06	ยืมหนังสือซ้ำ	book.borrow(); book.borrow();	Throw IllegalStateException	ระบบแจ้ง Exception ตามที่คาด ไม่มี Error อื่น	ผ่าน						

TC07	คืนหนังสือหลังเขียน	book.returnBook()	isBorrowed = false	ระบบเปลี่ยนสถานะ borrowed = false หลังการคืน ถูกต้องตามที่คาด	ผ่าน	
TC08	คืนหนังสือโดยไม่ได้เขียน	book.returnBook()	isBorrowed = false	ระบบขึ้งคงสถานะ borrowed = false และไม่เกิด Error ใด ๆ	ผ่าน	
TC09	เขียน—คืน—เขียนซ้ำ	เขียน > คืน > เขียน	สถานะเปลี่ยนถูกต้องทุกครั้ง	ระบบเปลี่ยนสถานะ borrowed ถูกต้องในทุกขั้นตอน (true → false → true)	ผ่าน	
TC10	author = null	("AI Basics", null)	title = "AI Basics", author = null	ระบบสร้างอีบอนเจกต์สำเร็จ โดย title = "AI Basics", author = null ไม่มี Error	ผ่าน	
TC11	เรียกคืนหนังสือซ้ำหลายครั้ง	book.returnBook() 2 ครั้ง	ไม่มี error	ระบบไม่เกิด Error เมื่อเรียก returnBook() หลายครั้งติดต่อกัน	ผ่าน	

โครงการ ระบบห้องสมุด (Library System)												
รหัสสถานการณ์การทดสอบ : SC02	ผู้เขียนสถานการณ์การทดสอบ : นางสาวเสาวลักษณ์ นุญรอด											
ฟังก์ชันที่ทดสอบ : Member() Member(String name), borrowBook(Book book), returnBook(Book book), getBorrowedBooks(), getName()	วันที่เขียน : 7 ตุลาคม 2568											
เงื่อนไขก่อนทดสอบ : มีการสร้างวัตถุ Member และ Book อย่างน้อย 1 ตัว เช่น Member m = new Member("Saowaluck"); Book b = new Book("AI Basics", "John Doe");												
เงื่อนไขหลังทดสอบ : สถานะของหนังสือและรายการที่สมาชิกเขียน (borrowedBooks) ถูกต้องตามการเพิ่มและลบ การคืนหนังสือหลายครั้งต้องไม่ทำให้เกิด Error												
ขั้นตอนการดำเนินการทดสอบ :												
1. ทดสอบการสร้างสมาชิกใหม่ด้วยชื่อปกติ												
2. ทดสอบชื่อสมาชิกเป็นค่าว่าง / null / ช่องว่าง												
3. ทดสอบการยืมหนังสือด้วย borrowBook()												
4. ทดสอบการ归还หนังสือด้วย returnBook()												
5. ทดสอบคืนหนังสือที่ไม่ได้เขียน (ไม่ควรเกิด error)												
6. ทดสอบการบันทึกหลายตัว และคืนเฉพาะบางตัว												
ทดสอบ getBorrowedBooks() ว่าสามารถแก้ไขจากภายนอกได้ (mutable)												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดขึ้นจริง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TC12	สร้างสมาชิกใหม่	"Saowaluck"	name ตรง, รายการเขียนว่าง	ระบบสร้างวัตถุ Member สำเร็จ ชื่อ และคงผลตรงตามค่าที่กำหนด และไม่มี	ผ่าน							

				รายการหนังสือใน borrowedBooks		
TC13	name ว่าง	""	Throw IllegalArgumentException	ระบบแจ้ง Exception IllegalArgumentException ตามที่คาด ไม่มี Error อื่นเกิดขึ้น	ผ่าน	
TC14	name = null	null	Throw IllegalArgumentException	ระบบแจ้ง Exception IllegalArgumentException ตรงตามที่คาด	ผ่าน	
TC15	name = ช่องว่าง	" "	Throw IllegalArgumentException	ระบบแจ้ง Exception IllegalArgumentException ถูกต้อง	ผ่าน	
TC16	ยืมหนังสือปกติ	member.borrowBook(book1)	book1 ถูกยืม, member มี 1 เล่ม	ระบบเพิ่ม book1 ลงใน borrowedBooks สำเร็จ และสถานะ book1 ถูก ตั้งเป็น borrowed = true	ผ่าน	
TC17	ยืมหนังสือ = null	member.borrowBook(null)	Throw IllegalArgumentException	ระบบตรวจสอบค่า null และโขน Exception ตามที่คาด ไม่มี Error อื่น	ผ่าน	
TC18	คืนหนังสือที่ยืม	member.returnBook(book1)	สามารถคืนได้	ระบบคืน book1 สำเร็จ และลบออกจาก borrowedBooks ถูกต้อง	ผ่าน	
TC19	คืนหนังสือที่ไม่ได้ยืม	member.returnBook(book1)	ไม่มี erro	ระบบไม่เกิด Error หรือ Exception เมื่อ book1 ไม่อยู่ในรายการ ยืม	ผ่าน	
TC20	ยืมหลายเล่ม	book1, book2	member.getBorrowedBooks().size() = 2	ระบบเพิ่มหนังสือทั้งสองเล่มลงใน borrowedBooks ได้ครบ และจำนวนเล่ม ถูกต้อง	ผ่าน	
TC21	ล้างรายการยืม	getBorrowedBooks().clear()	รายการว่าง	ระบบล้างรายการ borrowedBooks สำเร็จ ไม่มีข้อมูล ล้างใน list	ผ่าน	
TC22	คืน 1 จากหลายเล่ม	คืน book1 จาก book1, book2 ที่ยืม	book1 คืน, book2 ยังอยู่	ระบบลบเฉพาะ book1 ออกจาก borrowedBooks	ผ่าน	

				ถูกต้อง book2 ยังอยู่ในรายการ		
--	--	--	--	----------------------------------	--	--

โครงการ ระบบห้องสมุด (Library System)												
รหัสสถานการณ์การทดสอบ : SC03	ผู้เขียนสถานการณ์การทดสอบ : นางสาวสาวลักษณ์ นุญรอด วันที่เขียน : 7 ตุลาคม 2568											
ฟังก์ชันที่ทดสอบ : Loan() Loan(Book book, Member member), getBook(), getMember(), getBorrowDate(), getReturnDate(), returnBook()	เงื่อนไขก่อนทดสอบ : มีการสร้างวัตถุ Book และ Member แล้ว และหนังสือถูกยืม โดยสมาชิกนี้ เงื่อนไขหลังทดสอบ : เมื่อคืนหนังสือแล้ว ต้องตั้งค่า returnDate และ book.isBorrowed() กลับเป็น false หากคืนช้า ต้องไม่เกิดข้อผิดพลาด											
ขั้นตอนการดำเนินการทดสอบ :												
<ol style="list-style-type: none"> ทดสอบการสร้าง Loan ปกติ ทดสอบการสร้างโดยใช้ book หรือ member เป็น null ตรวจสอบค่า getBook(), getMember(), getBorrowDate() ว่าถูกต้อง ทดสอบ returnBook() และตรวจสอบว่าหนังสือถูกคืน ทดสอบ returnBook() ช้า (ต้องไม่ error) ทดสอบ returnBook() กับหนังสือที่ไม่ได้อยู่ในรายการยืมของสมาชิก 												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TC23	สร้าง Book ปกติ	title="Software Testing", author="Yogesh Singh"	สร้างสำเร็จ, isBorrowed=false	ระบบสร้างอ้อมจากค่า title ถูกต้อง มีค่า title และ author ตรงตามที่กำหนด และค่า isBorrowed = false	ผ่าน							
TC24	book=null	new Loan(null,member)	Exceptionถูกโyn	ระบบแจ้ง Exception ตามที่คาด (NullPointerException) ไม่มี Error อื่น	ผ่าน							
TC25	member=null	new Loan(book,null)	Exceptionถูกโyn	ระบบแจ้ง Exception ถูกต้อง (NullPointerException) และไม่สร้างอ้อมเขต Loan	ผ่าน							
TC26	คืนหนังสือปกติ	Loan.returnBook()	ReturnDate ถูกตั้งค่า	เมื่อเรียก Loan.returnBook() ระบบ เปลี่ยน ReturnDate เป็น วันที่ปัจจุบัน ถูกต้อง	ผ่าน							
TC27	คืนช้า	เรียก returnBook()ครั้ง ที่ 2	ไม่มีError, วันที่ไม่ เปลี่ยน	เมื่อเรียกคืนหนังสือช้า ระบบไม่เกิด Error และ ไม่เปลี่ยนวันที่เดิม	ผ่าน							

TC28	ตรวจสอบวันปีม	loan.getBorrowDate()	เท่ากับ LocalDate.now()	loan.getBorrowDate() คืน ค่าตรงกับ LocalDate.now() ในวันที่ ปีม	ผ่าน	
TC29	คืนหนังสือไม่อยู่ในmember list	bookใหม่	returnDate!=null	ระบบคืนค่า returnDate = null และไม่เกิด Error	ผ่าน	

โครงการ ระบบห้องสมุด (Library System)										
รหัสสถานการณ์ทดสอบ : SC04			ผู้เขียนสถานการณ์การทดสอบ : นางสาวสาวลักษณ์ บุญรอด							
ฟังก์ชันที่ทดสอบ : Library() Library(String name), addBook(Book book), borrowBook(String title, Member member), returnBook(Book book, Member member), getBooks(), getAvailableBookCount()			วันที่เขียน : 7 ตุลาคม 2568							
เงื่อนไขก่อนทดสอบ : มีการสร้างออบเจกต์ Library และเพิ่มหนังสือเข้าไปแล้ว เช่น library.addBook(new Book("Java Programming", "John Doe"))										
เงื่อนไขหลังทดสอบ : จำนวนหนังสือและสถานะของแต่ละเล่มในระบบต้องตรงตามผลการปีมและคืน										
ขั้นตอนการดำเนินการทดสอบ :										
<ol style="list-style-type: none"> ทดสอบการสร้าง Library ด้วยชื่อปกติทดสอบการสร้าง โดยใช้ book หรือ member เป็น null ตรวจสอบค่า getBook(), getMember(), getBorrowDate() ว่าถูกต้อง ทดสอบ returnBook() และตรวจสอบว่าหนังสือถูกคืน ทดสอบ returnBook() ช้า (ต้องไม่ error) ทดสอบ returnBook() กับหนังสือที่ไม่ได้อยู่ในรายการปีมของสมาชิก 										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TC30	สร้างLibrary	“Kasetsart Library”	ชื่อถูกต้อง,books ไม่ว่าง	ระบบสร้างออบเจกต์ “Kasetsart Library” และ books เริ่มต้น เป็นค่าว่าง	ผ่าน					
TC31	Name=NULL	””	Throw IllegalArgumentException	ระบบแข็ง Exception (IllegalArgumentException) ตามที่คาด ไม่มี Error อื่น	ผ่าน					
TC32	Name=NULL	null	Throw IllegalArgumentException	ระบบแข็ง Exception (IllegalArgumentException) และ ไม่สร้างออบเจกต์	ผ่าน					
TC33	Name=NULL	””	Throw IllegalArgumentException	ระบบแข็ง Exception (IllegalArgumentException) ข้าความที่คาด	ผ่าน					

TC34	เพิ่มหนังสือ	addBook(book)	มีหนังสือในlist	หลังเรียก addBook(book) หนังสือถูกเพิ่มใน list สำเร็จ	ผ่าน	
TC35	ขึ้นหนังสือสำเร็จ	borrowBook("Java Programming",member)	คืนค่า Loan.book.isBorrowed=true	ระบบคืนค่า Loan.book.isBorrowed = true หลังขึ้นสำเร็จ ไม่มี Error	ผ่าน	
TC36	ขึ้นหนังสือไม่พบ	borrowBook("No Book",member)	Throw IllegalArgumentException	ระบบแจ้ง Exception ประเภท IllegalArgumentException ถูกต้อง ไม่มี Error อื่น	ผ่าน	
TC37	ขึ้นช้า	ขึ้น2ครั้ง	Throw IllegalArgumentException	เมื่อขึ้นหนังสือที่ถูกขึ้น อยู่ ระบบแจ้ง Exception ประเภท IllegalArgumentException ตามที่คาด	ผ่าน	
TC38	คืนหนังสือสำเร็จ	returnBook(b2,member)	Book.isBorrowed=false	เมื่อเรียก returnBook(b2, member) ระบบตั้งค่า isBorrowed = false ถูกต้อง	ผ่าน	
TC39	นับจำนวนหนังสือ ที่ว่าง	getAvailableBookCount()	ตัวเลขถูกต้องทุก ขั้นตอน	เมื่อเรียก getAvailableBookCount() ระบบคืนค่า ตัวเลขถูกต้องทุก ขั้นตอน	ผ่าน	
TC40	ขึ้นทุกเดือน	borrowทั้งหมด	availableCount=0	เมื่อขึ้นหนังสือ ทั้งหมด ระบบตั้งค่า availableCount = 0 และ ไม่มี Error	ผ่าน	
TC41	getBooks() ปรับแต่งlist ได้	เพิ่มหรือลบในlist	จำนวนหนังสือเปลี่ยน ตาม	เมื่อเรียก getBooks() หลังเพิ่มหนังสือใหม่ จำนวนหนังสือใน list เพิ่มขึ้นตามจริง	ผ่าน	
TC42	ขึ้น-คืนหลายรอบ	Loop2รอบ	ไม่มีerror	ระบบทำงานถูกต้อง ในทุกครั้ง ไม่มี Error หรือค่าเพิ่ม	ผ่าน	

โครงการ ระบบห้องสมุด (Library System)											
รหัสสถานการณ์การทดสอบ : SC05		ผู้เขียนสถานการณ์การทดสอบ : นางสาวสาวลักษณ์ บุญรอด									
ฟังก์ชันที่ทดสอบ : Extra Coverage		วันที่เขียน : 7 ตุลาคม 2568									
เงื่อนไขก่อนทดสอบ : มีการสร้างอ็อบเจกต์ Library และเพิ่มหนังสือเข้าไปแล้ว เช่น library.addBook(new Book("Java Programming", "John Doe"))											
เงื่อนไขหลังทดสอบ : จำนวนหนังสือและสถานะของแต่ละเล่มในระบบต้องตรงตามผลการยืมและคืน											
ขั้นตอนการดำเนินการทดสอบ :											
<ol style="list-style-type: none"> ทดสอบการสร้าง Library ด้วยชื่อปกติทดสอบการสร้างโดยใช้ book หรือ member เป็น null ตรวจสอบค่า getBook(), getMember(), getBorrowDate() ว่าถูกต้อง ทดสอบ returnBook() และตรวจสอบว่าหนังสือถูกคืน ทดสอบ returnBook() ซ้ำ (ต้องไม่ error) ทดสอบ returnBook() กับหนังสือที่ไม่ได้อยู่ในรายการยืมของสมาชิก 											
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ					
TC43	ยืม-คืนซ้ำใน Library	ยืม Networks → คืน 2 ครั้ง	ไม่มี error	ระบบอนุญาตให้คืนหนังสือได้ปกติ และการคืนครั้งที่ 2 ไม่ทำให้เกิด Error หรือ Exception ใดๆ	ผ่าน						
TC44	Loan คืนซ้ำ	returnBook() 2 ครั้ง	returnDate ถูกกำหนด	ระบบคืนหนังสือสำเร็จ returnDate ถูกตั้งค่าอย่างถูกต้องตั้งเดิมรั้งแรก และไม่เปลี่ยนเมื่อคืนซ้ำ	ผ่าน						
TC45	Member คืนหนังสือที่ไม่เคยยืม	returnBook()	ไม่มี error	ระบบไม่เกิด Error หรือ Exception เมื่อหนังสือไม่ได้อยู่ในรายการยืมของสมาชิก	ผ่าน						
TC46	Library ลบหนังสือออกจาก list	remove(b)	รายการลดลง	ระบบลบหนังสือออกจากรายการได้สำเร็จ จำนวนหนังสือใน Library ลดลงตามคาด	ผ่าน						

โครงการ ระบบจัดการงาน (To-Do List System)											
รหัสสถานการณ์การทดสอบ : SC01		ผู้เขียนสถานการณ์การทดสอบ : นางสาวสาวลักษณ์ บุญรอด									
ฟังก์ชันที่ทดสอบ : Task()		วันที่เขียน : 7 ตุลาคม 2568									
เงื่อนไขก่อนทดสอบ : สร้างวัตถุ Task เรียบร้อย เช่น Task task = new Task("Do Homework");											
เงื่อนไขหลังทดสอบ : ค่าของ isDone เป็นเท็จแปลงถูกต้อง, ไม่มีการโยนข้อผิดพลาดที่ไม่คาดคิด											
ขั้นตอนการดำเนินการทดสอบ :											
<ol style="list-style-type: none"> ทดสอบการสร้าง Library ด้วยชื่อปกติทดสอบการสร้าง โดยใช้ book หรือ member เป็น null ตรวจสอบค่า getBook(), getMember(), getBorrowDate() ว่าถูกต้อง ทดสอบ returnBook() และตรวจสอบว่าหนังสือถูกคืน ทดสอบ returnBook() ช้า (ต้องไม่ error) ทดสอบ returnBook() กับหนังสือที่ไม่ได้อยู่ในรายการยืมของสมาชิก 											
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ					
TC01	สร้าง Task ใหม่	"Do Homework"	title = "Do Homework", isDone = false	ระบบสร้างอื่น เล็กต์สำเร็จ ค่าชื่อและสถานะตรงตามคาด (isDone = false)	ผ่าน	ตรวจสอบ getTitle(), isDone()					
TC02	title ว่าง	""	Throw IllegalArgumentException	ระบบแจ้ง Exception ตามที่คาด ไม่มี Error อื่น	ผ่าน	ป้องกันข้อมูลไม่ถูกต้อง					
TC03	title = null	null	Throw IllegalArgumentException	ระบบโยน Exception ถูกต้อง (IllegalArgumentException)	ผ่าน						
TC04	title = ช่องว่าง	""	Throw IllegalArgumentException	ระบบแจ้ง Exception ถูกต้อง ไม่มี Error อื่น	ผ่าน						
TC05	markDone → markUndone	เรียก markDone(), markUndone()	สถานะเปลี่ยน true → false	ระบบเปลี่ยนสถานะจาก isDone = true กลับเป็น false ถูกต้อง	ผ่าน						

TC06	toggle หลายรอบ	เรียกสลับ 3 ครั้ง	สถานะเปลี่ยนถูกต้องทุกครั้ง	สถานะกลับไปกลับระหว่าง true และ false ได้ถูกต้องทุกครั้ง	ผ่าน	
TC07	ทดสอบหา Task พร้อมกัน	"Read Book", "Do Exercise"	markDone เฉพาะ t1, t2 ยังไม่ทำ	t1 แสดง isDone = true, t2 ยัง isDone = false	ผ่าน	

โครงการ ระบบจัดการงาน (To-Do List System)										
รหัสสถานการณ์ทดสอบ : SC02			ผู้เขียนสถานการณ์ทดสอบ : นางสาวสาวลักษณ์ บุญรอด							
ฟังก์ชันที่ทดสอบ : TaskManager () TaskManager(), addUser(User user), findUser(String name)			วันที่เขียน : 7 ตุลาคม 2568							
เงื่อนไขก่อนทดสอบ : มีการสร้างตัวอย่าง TaskManager แล้ว และ User หลายคนพร้อมใช้ทดสอบ										
เงื่อนไขหลังทดสอบ : User ต้องถูกเพิ่ม/คืนหากได้ถูกต้องตามข้อที่กำหนด หากข้อซ้ำ ควรไม่เกิดข้อผิดพลาด										
ขั้นตอนการดำเนินการทดสอบ :										
<ol style="list-style-type: none"> ทดสอบการสร้าง Library ด้วยชื่อปกติทดสอบการสร้าง โดยใช้ book หรือ member เป็น null ตรวจสอบค่า getBook(), getMember(), getBorrowDate() ว่าถูกต้อง ทดสอบ returnBook() และตรวจสอบว่าหนังสือถูกคืน ทดสอบ returnBook() ซ้ำ (ต้องไม่ error) ทดสอบ returnBook() กับหนังสือที่ไม่ได้อยู่ในรายการยืมของสมาชิก 										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TC08	เพิ่ม User ใหม่	addUser("Saowaluck")	findUser("Saowaluck") != null	ระบบเพิ่มผู้ใช้ใหม่ได้สำเร็จ เมื่อเรียก findUser("Saowaluck") คืนค่าอื่นจากค่าใช้ที่ตรงซึ่ง	ผ่าน					
TC09	เพิ่ม User = null	addUser(null)	Throw IllegalArgumentExceptionException (IllegalArgumentExceptionException) ตามที่คาด	ระบบตรวจสอบอินพุตและแจ้ง Exception (IllegalArgumentExceptionException) ตามที่คาด	ผ่าน	ป้องกันข้อมูลไม่ถูกต้อง				
TC10	คืนหน้าผู้ใช้	findUser("Naramon Sirakun")	คืนค่า User ที่ตรงซึ่ง	ระบบคืนหน้าและคืนอีบเจกต์ผู้ใช้ "Naramon Sirakun" ถูกต้อง	ผ่าน					

TC11	ค้นหาไม่พบ	findUser("Nisarat Kawpok")	คืนค่า null	ระบบคืนค่า null เมื่อไม่พบชื่อผู้ใช้ในรายการ	ผ่าน	
TC12	เพิ่มชื่อช้า	addUser("Saowaluck") ช้า	findUser() ยังคืนได้	ระบบไม่เกิด Error เมื่อเพิ่มชื่อช้า ผู้ใช้เก่าคงอยู่ findUser("Saowaluck") ยังคืนค่าได้ถูกต้อง	ผ่าน	

โครงการ ระบบจัดการงาน (To-Do List System)										
รหัสสถานการณ์ทดสอบ : SC03			ผู้เขียนสถานการณ์ทดสอบ : นางสาวสาวลักษณ์ บุญรอด							
ฟังก์ชันที่ทดสอบ : User () User(String name), getName(), addTask(Task task), removeTask(String title), getTasks()			วันที่เขียน : 7 ตุลาคม 2568							
เงื่อนไขก่อนทดสอบ : มีการสร้างวัตถุ User และ Task อ yogurt 1 รายการแล้ว										
เงื่อนไขหลังทดสอบ : จำนวน Task ที่ผู้ใช้ดึงข้อมูลต้องตรงกับผลการเพิ่ม/ลบ ไม่มี Exception ที่ไม่คาดคิด										
ขั้นตอนการดำเนินการทดสอบ :										
<ol style="list-style-type: none"> ทดสอบการสร้าง Library ด้วยชื่อปกติทดสอบการสร้าง โดยใช้ book หรือ member เป็น null ตรวจสอบค่า getName(), getMember(), getBorrowDate() ว่าถูกต้อง ทดสอบ returnBook() และตรวจสอบว่าหนังสือถูกคืน ทดสอบ returnBook() ช้า (ต้องไม่ error) ทดสอบ returnBook() กับหนังสือที่ไม่ได้อยู่ในรายการยืมของสมาชิก 										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TC13	สร้าง User ใหม่	addUser("Saowaluck")	name = "Saowaluck", tasks = []	ระบบสร้าง User สำเร็จ getName() คืนค่า "Saowaluck" และ getTasks() คืนค่า list ว่าง []	ผ่าน	ตรวจสอบ getName(), getTasks()				
TC14	name ว่าง	""	Throw IllegalArgumentException	ระบบโYN IllegalArgumentException ตามที่คาด	ผ่าน	ป้องกัน ข้อมูลไม่ถูกต้อง				
TC15	name = null	null	Throw IllegalArgumentException	ระบบตรวจสอบและ โYN IllegalArgumentException ได้ถูกต้อง	ผ่าน					
TC16	เพิ่มงาน 1 งาน	addTask("Do Homework")	จำนวน = 1, task ตรง	ระบบเพิ่ม task ให้สำเร็จ getTasks() มี 1 รายการ ชื่อ "Do Homework"	ผ่าน					

TC17	เพิ่มงาน = null	addTask(null)	Throw IllegalArgument Exception	ระบบตรวจสอบว่า task คือ null และแจ้ง Exception ตามที่คาด	ผ่าน	
TC18	ลบงานปกติ	removeTask("Read Exam Book")	task ถูกลบ, เหลือ 1	ระบบลบ task "Read Exam Book" ออกสำเร็จ getTasks() เหลือ 1 รายการ	ผ่าน	
TC19	ลบงานไม่พบ	removeTask("Cooking")	ไม่มี error, จำนวนเท่าเดิม	ระบบไม่พบงานชื่อ "Cooking" แต่ไม่เกิด Error และจำนวน task เท่าเดิม	ผ่าน	
TC20	ลบงานไม่สนใจตัวพิมพ์	removeTask("read exam book")	ลบสำเร็จ	ระบบสามารถลบงาน "read exam book" ได้แม้พิมพ์เล็ก-ใหญ่ต่างกัน (case-insensitive)	ผ่าน	
TC21	ลบหลายงาน	removeTask("Do Homework"), removeTask("Go Running")	เหลือ 1 งาน	ระบบลบงานทั้งสองสำเร็จ getTasks() เหลือเพียง 1 รายการ สุดท้าย	ผ่าน	
TC22	getTasks() เป็น mutable list	แก้ไข list ภายนอก	ขนาด list เปลี่ยน	เมื่อแก้ไข list ที่ได้จาก getTasks() ข้อมูล task ของ User จะเปลี่ยนตามจริง (พิสูจน์ว่า list เป็น mutable)	ผ่าน	
TC23	ลบงานทั้งหมด	clear()	รายการว่างเปล่า	หลังรัน clear() รายการ task ใน User เป็น list ว่าง []	ผ่าน	

โครงการ API Testing – Fake Store API						
รหัสสถานการณ์การทดสอบ : SC01	ผู้เขียนสถานการณ์การทดสอบ : นางสาววนิดา ศิลาภูล					
พังก์ชันที่ทดสอบ :	วันที่เขียน : 10 ตุลาคม 2568					
การทดสอบ API ด้วย HTTP Methods:	<ul style="list-style-type: none"> - GET (Get a Single Product) <p>โดยใช้เครื่องมือ Postman และ Katalon Studio</p> <p>เว็บไซต์ที่ใช้ทดสอบ:</p> <p>https://fakestoreapi.com/docs#tag/Products</p>					
เงื่อนไขก่อนทดสอบ :	<ul style="list-style-type: none"> - ใช้ Postman สำหรับทดสอบแบบ Manual API Testing - ใช้ Katalon Studio สำหรับทดสอบแบบ Automation API Testing - กำหนด Environment API เป็น https://fakestoreapi.com 					
เงื่อนไขหลังทดสอบ :	<ul style="list-style-type: none"> - API ตอบกลับสำเร็จ (Status Code 200 OK) - ระบบสามารถดึงข้อมูลสินค้า ID ที่กำหนดได้จริง - แสดงค่าพิเศษของทุกตัว ได้แก่ id, title, price, description, category, image 					
ขั้นตอนการดำเนินการทดสอบ :	<ul style="list-style-type: none"> - เปิดโปรแกรม Postman - สร้าง Collection ชื่อ “PROJECT” และเพิ่ม Request ดังนี้ <ul style="list-style-type: none"> - Get_Product_Success → ทดสอบเรียกคืนค่ารายตัว - Get_Product_InvalidID → ทดสอบเรียกคืนค่าที่ไม่มีอยู่ - กำหนด Body, URL, Header ตาม API Documentation ของ FakeStoreAPI - กด Send เพื่อทดสอบแต่ละ Request - ตรวจสอบ Status Code และ Response Body ที่ได้จาก API - บันทึกผลการทดสอบในรายงาน (ส่วนตารางด้านล่าง) - ทำซ้ำขั้นตอนใน Katalon Studio โดยสร้าง Test Case เดียวกันในรูปแบบ Automation - รัน Test Suite และบันทึกผลลัพธ์จาก Log Viewer 					
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
TC_GET_01	ดึงข้อมูลสินค้ารายตัว (Single Product)	URL: GET /products/1	ได้ Status Code 200 OK และ Response แสดงข้อมูลสินค้าที่มี id=1 (title, price, description, category, image)	ได้ Status Code 200 OK พร้อมข้อมูลสินค้าครบ	ผ่าน	
TC_GET_02	ดึงข้อมูลสินค้าด้วย ID ที่ไม่มีอยู่ (Invalid ID)	URL: GET /products/999	ได้ Status Code 404 Not Found หรือไม่มีข้อมูล	ได้ Status Code 200 OK แต่ Response เป็นค่าว่าง	ผ่าน	FakeStoreAPI ไม่ตรวจสอบ ID จริง

โครงการ API Testing – Fake Store API						
รหัสสถานการณ์ทดสอบ : SC02		ผู้เขียนสถานการณ์ทดสอบ : นางสาวนรรณ ศิลาภูมิ				
พื้นที่ที่ทดสอบ :						
การทดสอบ API ด้วย HTTP Methods:						
<ul style="list-style-type: none"> - POST (Add a New Product) 	วันที่เขียน : 10 ตุลาคม 2568					
โดยใช้เครื่องมือ Postman และ Katalon Studio						
เครื่องมือที่ใช้ทดสอบ:						
https://fakestoreapi.com/docs#tag/Products						
เงื่อนไขก่อนทดสอบ :						
<ul style="list-style-type: none"> - ใช้ Postman สำหรับทดสอบแบบ Manual API Testing - ใช้ Katalon Studio สำหรับทดสอบแบบ Automation API Testing - กำหนด Environment API เป็น https://fakestoreapi.com 						
เงื่อนไขหลังทดสอบ :						
<ul style="list-style-type: none"> - ระบบตอบกลับด้วย Status Code 201 Created - มีการสร้างสินค้าใหม่สำเร็จ และ Response แสดงข้อมูลสินค้าที่เพิ่ม (มี id ใหม่ในระบบ) - ระบบสามารถคืนค่าข้อมูลสินค้าที่สร้างใหม่ได้ถูกต้อง 						
ขั้นตอนการดำเนินการทดสอบ :						
<ul style="list-style-type: none"> - เปิดโปรแกรม Postman - สร้าง Collection ชื่อ “PROJECT” และเพิ่ม Request ดังนี้ <ul style="list-style-type: none"> - Post_Product_Success → ทดสอบเพิ่มสินค้าใหม่ - Post_Product_MissingData → ทดสอบเพิ่มสินค้าข้อมูลไม่ครบ - กำหนด Body, URL, Header ตาม API Documentation ของ FakeStoreAPI - กด Send เพื่อทดสอบแต่ละ Request - ตรวจสอบ Status Code และ Response Body ที่ได้จาก API - บันทึกผลการทดสอบในรายงาน (ส่วนตารางด้านล่าง) - ทำซ้ำขั้นตอนใน Katalon Studio โดยสร้าง Test Case เดียวกันในรูปแบบ Automation - รัน Test Suite และบันทึกผลลัพธ์จาก Log Viewer 						
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
TC_POST_01	เพิ่มสินค้าใหม่	URL: POST /products Body: <pre>{"title":"New Test Product","price":13.5,"description":"A Great Product","image":"https://i.pravatar.cc","category":"Electronics"}</pre>	ได้ Status Code 201 Created และ Response แสดงข้อมูลสินค้าที่เพิ่มใหม่	ได้ Status Code 201 Created และ Response แสดงข้อมูลครบ	ผ่าน	
TC_POST_02	เพิ่มสินค้าข้อมูลไม่ครบ	URL: POST /products Body:	API ควรตอบกลับด้วย Status Code 200 หรือ 201 (ถือว่าล้มเหลว เมื่อข้อมูลไม่ครบ)	API ตอบกลับ Status Code 200 มี title อยู่ใน	ผ่าน	ปรับให้ดีใหม่ให้ตรวจสอบฟิลด์ title, price ได้

	(Missing Data)	json { "title": "Product Missing Data" }	ไม่มีค่าในฟิลด์ price หรือ category และ title เป็น null หรือไม่มีค่า	response และฟิลด์ price, category เป็น null หรือไม่ปรากฏ ระบบตรวจสอบผ่านเงื่อนไขทั้งหมด		ถูกต้อง และผ่านการทดสอบ
--	----------------	--	--	--	--	-------------------------

โครงการ API Testing – Fake Store API									
รหัสสถานการณ์การทดสอบ : SC03			ผู้เขียนสถานการณ์การทดสอบ : นางสาววนรุณ พิลาภุล						
พังก์ชันที่ทดสอบ :				วันที่เขียน : 10 พฤษภาคม 2568					
การทดสอบ API ด้วย HTTP Methods:									
<ul style="list-style-type: none"> - PUT (Update a Product) <p>โดยใช้เครื่องมือ Postman และ Katalon Studio</p> <p>เว็บไซต์ที่ใช้ทดสอบ:</p> <p>https://fakestoreapi.com/docs#tag/Products</p>									
เงื่อนไขก่อนทดสอบ :									
<ul style="list-style-type: none"> - ใช้ Postman สำหรับทดสอบแบบ Manual API Testing - ใช้ Katalon Studio สำหรับทดสอบแบบ Automation API Testing - กำหนด Environment API เป็น https://fakestoreapi.com 									
เงื่อนไขหลังทดสอบ :									
<ul style="list-style-type: none"> - ระบบตอบกลับด้วย Status Code 200 OK - Response แสดงข้อมูลเดิมคืนที่ถูกอัปเดต เช่น title, price, description, category - ไม่มี Error และข้อมูลอัปเดตตรงตามที่ส่งไป 									
ขั้นตอนการดำเนินการทดสอบ :									
<ul style="list-style-type: none"> - เปิดโปรแกรม Postman - สร้าง Collection ชื่อ “PROJECT” และเพิ่ม Request ดังนี้ <ul style="list-style-type: none"> - Put_Product_Success → ทดสอบอัปเดตข้อมูลเดิมคืน - กำหนด Body, URL, Header ตาม API Documentation ของ FakeStoreAPI - กด Send เพื่อทดสอบแต่ละ Request - ตรวจสอบ Status Code และ Response Body ที่ได้จาก API - บันทึกผลการทดสอบในรายงาน (ส่วนตารางค้านล่าง) - ทำข้อความใน Katalon Studio โดยสร้าง Test Case เดียวกันในรูปแบบ Automation - รัน Test Suite และบันทึกผลลัพธ์จาก Log Viewer 									
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ			
TC_PUT_01	อัปเดตข้อมูลเดิม	URL: PUT /products/1 Body: {"id":1,"title":"Updated Test Product","price":99.99, "category":"Electronics","rating":5.0,"stock":100}	ได้ Status Code 200 OK และ Response แสดงข้อมูลที่อัปเดต	ได้ Status Code 200 OK และ Response แสดงข้อมูลที่แก้ไขแล้ว	ผ่าน				

		"description":"Updated description for test product","image":"https://i.pra vatar.cc","category":"electroni cs"}			
--	--	--	--	--	--

โครงการ API Testing – Fake Store API	
รหัสสถานการณ์การทดสอบ : SC04	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนรรณ ศิลาภูมิ
พื้นที่ที่ทดสอบ : การทดสอบ API ด้วย HTTP Methods: - DELETE (Delete a Product) โดยใช้เครื่องมือ Postman และ Katalon Studio เว็บไซต์ที่ใช้ทดสอบ: https://fakestoreapi.com/docs#tag/Products	วันที่เขียน : 10 ตุลาคม 2568

เงื่อนไขก่อนทดสอบ :
- ระบบตอบกลับด้วย Status Code 200 OK
- Response แสดงข้อมูลสินค้าที่ถูกลบออก
- ไม่มี Error หรือ Response ที่ผิดพลาด
- สินค้าที่ถูกลบไม่ถูกเรียกซ้ำในรายการ

ขั้นตอนการดำเนินการทดสอบ :
- เปิดโปรแกรม Postman
- สร้าง Collection ชื่อ “PROJECT” และเพิ่ม Request ดังนี้
- Delete_Product_Success → ทดสอบลบสินค้า
- กำหนด Body, URL, Header ตาม API Documentation ของ FakeStoreAPI
- กด Send เพื่อทดสอบแต่ละ Request
- ตรวจสอบ Status Code และ Response Body ที่ได้จาก API
- บันทึกผลการทดสอบในรายงาน (ส่วนตารางด้านล่าง)
- ทำซ้ำขั้นตอนใน Katalon Studio โดยสร้าง Test Case เดียวกันในรูปแบบ Automation
- รัน Test Suite และบันทึกผลลัพธ์จาก Log Viewer

รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
TC_DELETE_01	ลบสินค้าสำเร็จ	URL: DELETE /products/1	ได้ Status Code 200 OK และ Response แสดงข้อมูลสินค้าที่ถูกลบ	ได้ Status Code 200 OK พร้อมข้อมูลสินค้า id=1	ผ่าน	

--	--	--	--	--	--	--

โครงการ : XYZ Bank												
รหัสสถานการณ์การทดสอบ : SC01	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก											
ฟังก์ชันที่ทดสอบ : Customer Login	วันที่เขียน : 12 ตุลาคม 2568											
เดือนที่ทดสอบ : ผู้ใช้คลิกลิงก์ https://www.globalsqa.com/angularJs-protractor/BankingProject												
เงื่อนไขทดสอบ :												
ขั้นตอนการดำเนินการทดสอบ : ผู้ใช้คลิกลิงก์ https://www.globalsqa.com/angularJs-protractor/BankingProject												
- ผู้ใช้เลือก Customer Login												
- ผู้ใช้เลือกชื่อรายการและการกดปุ่ม Login ข้ามสู่ระบบ												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TC01	ผู้ใช้ Login สำเร็จ	ผู้ใช้คลิกปุ่ม Customer Login, ผู้ใช้เลือกข้อมูล ชื่อจากรายการ และกดปุ่ม Login	ไปหน้ารายการ ธุรกรรม	ไปหน้ารายการ ธุรกรรม	ผ่าน							

โครงการ : XYZ Bank												
รหัสสถานการณ์การทดสอบ : SC02	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก											
ฟังก์ชันที่ทดสอบ : Bankmanager Login	วันที่เขียน: 12 ตุลาคม 2568											
เดือนที่ทดสอบ : ผู้ใช้คลิกลิงก์ https://www.globalsqa.com/angularJs-protractor/BankingProject												
เงื่อนไขทดสอบ :												
ขั้นตอนการดำเนินการทดสอบ : ผู้ใช้คลิกลิงก์ https://www.globalsqa.com/angularJs-protractor/BankingProject												
- ผู้ใช้เลือกBank manager Login												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TC02	ผู้ใช้ Login สำเร็จ	ผู้ใช้คลิกปุ่ม Bankmanager Login	แสดงหน้ารายการ	แสดงหน้ารายการ	ผ่าน							

โครงการ :XYZ Bank											
รหัสสถานการณ์การทดสอบ : SC03		ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก									
พื้นที่ที่ทดสอบ : Deposit		วันที่เขียน: 12 ตุลาคม 2568									
เงื่อนไขก่อนทดสอบ : -Customer Login เข้าสู่ระบบสำเร็จ											
เงื่อนไขหลังทดสอบ :											
ขั้นตอนการดำเนินการทดสอบ - ผู้ใช้คลิกลิงก์ https://www.globalsqa.com/angularJs-protractor/BankingProject											
:ผู้ใช้เลือกCustomer Login											
-ผู้ใช้ทำการLoginเข้าสู่ระบบ											
-ผู้ใช้เลือกรายการDeposit											
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ					
TC03	ผู้ใช้ทำการฝากเงินสำเร็จ	ผู้ใช้กรอกจำนวนเงิน"1,000"	แสดงข้อความ"Deposit Successful"และแสดงยอดเงินรวม	แสดงข้อความ"Deposit Successful"และยอดเงินรวม	ผ่าน						
TC04	ผู้ใช้กรอกข้อมูลไม่ถูกต้อง	ผู้ใช้กรอกจำนวนเงิน"0"	แสดงข้อความ"Deposit Failed"	ไม่สามารถกดปุ่ม Deposit ได้	ไม่ผ่าน						

โครงการ :XYZ Bank	
รหัสสถานการณ์การทดสอบ : SC04	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก
ฟังก์ชันที่ทดสอบ : Withdrawl	วันที่เขียน: 12 ตุลาคม 2568
เงื่อนไขก่อนทดสอบ : ผู้ใช้ทำการฝากเงินสำเร็จ	
เงื่อนไขหลังทดสอบ :	
ขั้นตอนการดำเนินการทดสอบ; -ผู้ใช้คลิกลิงก์ https://www.globalsqa.com/angularJs-protractor/BankingProject	
:ผู้ใช้เลือกCustomer Login	
-ผู้ใช้ทำการLoginเข้าสู่ระบบ	
-ผู้ใช้เลือกรายการDeposit	
-ผู้ใช้ทำการฝากเงินสำเร็จ	
-ผู้ใช้เลือกรายการWithdrawl	
-ผู้ใช้กรอกจำนวนเงินและกดปุ่มWithdrawl	

รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
TC05	ผู้ใช้ทำการถอนเงินสำเร็จ	ผู้ใช้กรอก จำนวน เงิน”100”	แสดง ข้อความ”Withdrawl Successful”ระบบทำ การหักยอดเงินและ แสดงยอดเงินคงเหลือ	แสดง ข้อความ”Transaction Successful”ระบบทำการ หักยอดเงินและแสดง ยอดเงินคงเหลือ	ไม่ผ่าน	
TC06	ผู้ใช้กรอก ข้อมูลไม่ ถูกต้อง	ผู้ใช้กรอก จำนวนเงิน”0”	แสดงข้อความ”Withd rawFailed”	ไม่สามารถกดคปุ่ม Withdrawlได้	ไม่ผ่าน	
TC07	ผู้ใช้กรอก จำนวนเงิน มากกว่าจำนวน เงินที่ฝาก	ผู้ใช้กรอก จำนวน เงิน”6,000”	แสดงข้อความแจ้ง เตือน” Transaction Failed”	แสดงข้อความแจ้งเตือน” Transaction Failed. You can not withdraw amount more than the balance”	ไม่ผ่าน	

โครงการ :XYZ Bank	
รหัสสถานการณ์การทดสอบ : SC05	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก
ฟังก์ชันที่ทดสอบ : Transactions	วันที่เขียน: 12 ตุลาคม 2568
เงื่อนไขก่อนทดสอบ : ผู้ใช้ทำการฝากเงินหรือถอนเงินสำเร็จ	
เงื่อนไขหลังทดสอบ :	
ขั้นตอนการดำเนินการทดสอบ :	
-ผู้ใช้คลิกลิงก์ https://www.globalsqa.com/angularJs-protractor/BankingProject	
:ผู้ใช้เลือกCustomer Login	
-ผู้ใช้ทำการLoginเข้าสู่ระบบ	
-ผู้ใช้เลือกรายการDepositหรือWithdrawl	
-ผู้ใช้ทำการสำเร็จ	
-ผู้ใช้เลือกรายการTransactions	

รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
TC08	ผู้ใช้ชื่อประวัติ ธุรกรรม	ผู้ใช้คลิกปุ่ม Transactions	แสดงรายการ วัน เดือน ปี เวลา ที่ทำการ แสดงยอดเงินและ ประเภทธุรกรรม	แสดงรายการ วัน เดือน ปี เวลา ที่ทำการ แสดงยอดเงินและ ประเภทธุรกรรม	ผ่าน	
TC09	ผู้ใช้ทำการ Resetข้อมูล	ผู้ใช้คลิกปุ่ม Reset	ระบบทำการลบประวัติ การทำการธุรกรรม ออกทั้งหมด	ระบบทำการลบประวัติ การทำการธุรกรรม ออกทั้งหมด	ผ่าน	

โครงการ :XYZ Bank	
รหัสสถานการณ์การทดสอบ : SC06	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสรารัตน์ แก้วปีอก
ฟังก์ชันที่ทดสอบ : Add Customer	วันที่เขียน: 12 ตุลาคม 2568
เงื่อนไขก่อนทดสอบ : Bank Manager Login สำหรับเจ้าหน้าที่	

เงื่อนไขหลังทดสอบ

ที่นี่ตอกย้ำการดำเนินการทดสอบ - ผู้ใช้คือกิจลักษณ์ <https://www.globalsqa.com/angularJs-protractor/BankingProject>

· ผู้ใช้เลือก Bank Manager Login

-ผู้ใช้ทำการ Log in เข้าสู่ระบบ

-**ผู้ใช้** เลือก Add Customer

- ฝึกใช้กรอกข้อมูลชื่อ นามสกุล รหัส และกดปุ่ม Add Customer

รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
TC10	ผู้ใช้ทำการเพิ่ม ข้อมูลลูกค้า สำเร็จ	ผู้ใช้กรอกข้อมูล FirstName:”Obanai “ LastName:”iguro” PostCode:1001	แสดง ข้อความ”Customer Add Successful”	มีแท็บแจ้ง เตือน”customer added successfully with customer id:6”	ไม่ผ่าน	
TC11	ผู้ใช้กรอก ข้อมูลไม่ครบ	ผู้ใช้กรอกข้อมูล FirstName:”Obanai “ LastName:Null PostCode:1001	แสดงข้อความ” กรุณากรอกข้อมูล ในช่องนี้”	แสดงข้อความแจ้ง เตือน”โปรดกรอก ข้อมูลฟิลด์นี้”	ไม่ผ่าน	

โครงการ :XYZ Bank	
รหัสสถานการณ์การทดสอบ : SC07	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก
ฟังก์ชันที่ทดสอบ : Open Account	วันที่เขียน: 12 ตุลาคม 2568
เงื่อนไขก่อนทดสอบ : Bank Manager Login สำหรับเจ้าหน้าที่	
เงื่อนไขหลังทดสอบ :	
ขั้นตอนการดำเนินการทดสอบ - คลิกลิงค์ https://www.globalsqa.com/angularJs-protractor/BankingProject	

<p>:ผู้ใช้เลือกCustomer Login</p> <p>-ผู้ใช้ทำการLoginเข้าสู่ระบบ</p> <p>-ผู้ใช้เลือกรายการOpen Account</p>						
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ
TC12	ผู้ใช้เป็นบัญชีสำเร็จ	ผู้ใช้เลือกชื่อจากรายการ"Ron Wesly" เลือกCurrencyจากรายการ"Dollar"	แสดงข้อความแจ้งเตือน"Account Created Successfully"	มีแท็บแจ้งเตือน"Account Created Successfully With account Number :1016"	"ไม่ผ่าน"	
TC13	ผู้ใช้เลือกรายการข้อมูลไม่ครบ	ผู้ใช้เลือกชื่อจากรายการ"Albus Dumbledore" ไม่ทำการเลือกรายการCurrency	แสดงข้อความ"กรุณาเลือกรายการ"	แสดงข้อความ"กรุณาเลือกรายการ"	"ผ่าน"	

โครงการ :XYZ Bank										
รหัสสถานการณ์การทดสอบ : SC08			ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก							
ฟังก์ชันที่ทดสอบ : Logout			วันที่เขียน: 12 ตุลาคม 2568							
เงื่อนไขก่อนทดสอบ : Customer Loginเข้าสู่ระบบสำเร็จ										
เงื่อนไขหลังทดสอบ :										
ขั้นตอนการดำเนินการทดสอบ -ผู้ใช้คลิกลิงก์ https://www.globalsqa.com/angularJs-protractor/BankingProject										
-ผู้ใช้ทำการLoginเข้าสู่ระบบ										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TC14	ผู้ใช้ทำการLogout	ผู้ใช้คลิกปุ่มLogout	กลับไปหน้า Home Page	กลับไปหน้ากรอกข้อมูลชื่อ	"ไม่ผ่าน"					

โครงการ :Katalon Shop										
รหัสสถานการณ์การทดสอบ : SC01			ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก							
ฟังก์ชันที่ทดสอบ : Login			วันที่เขียน : 13 ตุลาคม 2568							
เงื่อนไขก่อนทดสอบ : ผู้ใช้มีบัญชีอยู่แล้ว										
เงื่อนไขหลังทดสอบ : อยู่ในหน้า Dashboard										
ขั้นตอนการดำเนินการทดสอบ : ผู้ใช้คลิกลิงก์ Katalon Shop – Katalon Ecommerce										
-ผู้ใช้เลือก My Account										
-ผู้ใช้กรอกข้อมูล Username และ Password										
-ผู้ใช้คลิกปุ่มLogin										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				

TC01	ผู้ใช้ทำการ Loginเข้าสู่ระบบสำเร็จ	Username:"customer" Password:crz7mrb.KNG3yxv1fbn	แสดงข้อความแจ้งเตือน"Login Success"	หน้าจอแสดงข้อความHello(ชื่อบัญชีผู้ใช้)และแสดงปุ่มLogout,Recentorders,Shipping and billing address!! และ edit your password and account details	ไม่ผ่าน	
TC02	ผู้ใช้กรอกข้อมูลผิด	Username:"customer" Password:1234	แสดงข้อความ"username or password is incorrect"	แสดงข้อความ"The password you entered for the username is incorrect.Lost your password	ไม่ผ่าน	

โครงการ:Katalon Shop												
รหัสสถานการณ์ทดสอบ : SC02	ผู้ใช้ยืนยันการตั้งค่าการทดสอบ : นางสาวนิสารัตน์ แก้วปือก											
ฟังก์ชันที่ทดสอบ : ADD TO CART	วันที่เขียน : 13 ตุลาคม 2568											
เงื่อนไขก่อนทดสอบ : ผู้ใช้ทำการLogin สำเร็จ												
เงื่อนไขหลังทดสอบ : มีสินค้าอยู่ในรถเข็น												
ขั้นตอนการดำเนินการทดสอบ : -ผู้ใช้คลิกลิงก์ Katalon Shop – Katalon Ecommerce												
-ผู้ใช้ทำการLoginเข้าสู่ระบบสำเร็จ												
-ผู้ใช้เลือกรายการShop												
-ผู้ใช้เลือกรายการสินค้าที่ต้องการ												
-ผู้ใช้กดปุ่มADD TO CARTตรงสินค้าที่ต้องการเพิ่ม												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TC03	ผู้ใช้เพิ่มสินค้าลงรายการรถเข็นสำเร็จ	ผู้ใช้คลิกปุ่มADD TO CART	แสดงข้อความ"สินค้าถูกเพิ่มลงรายการ"	แสดงข้อความ"(ชื่อสินค้า) has been added in your cart"	ไม่ผ่าน							

โครงการ :Katalon Shop												
รหัสสถานการณ์การทดสอบ : SC03	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปือก											
ฟังก์ชันที่ทดสอบ : CART	วันที่เขียน : 10 ตุลาคม 2568											
เงื่อนไขก่อนทดสอบ : เพิ่มสินค้าลงในรายการรถเข็นสำเร็จ												
เงื่อนไขหลังทดสอบ :												
ขั้นตอนการดำเนินการทดสอบ : ผู้ใช้คลิกลิงก์ Katalon Shop – Katalon Ecommerce												
<ul style="list-style-type: none"> - ผู้ใช้ทำการ Login เข้าสู่ระบบสำเร็จ - ผู้ใช้เลือกรายการ Shop - ผู้ใช้กดปุ่ม ADD TO CART ของสินค้าที่ต้องการเพิ่ม - ผู้ใช้พิมพ์สินค้าลงรายการรถเข็นสำเร็จ - ผู้ใช้เลือกรายการ CART เพื่อดูรายการสินค้า - ผู้ใช้กดปุ่ม PROCEED TO CHECK OUT 												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TC04	ผู้ใช้กรอกรายการสินค้าที่เพิ่มลงรถเข็น	ผู้ใช้คลิกเลือกรายการ CART	หน้าจอแสดงรายการสินค้าที่ทำการเพิ่ม	หน้าจอแสดงรายการสินค้าที่ทำการเพิ่ม	ผ่าน							
TC05	ผู้ใช้ลบรายการสินค้า	ผู้ใช้คลิกที่เครื่องหมายกาหนาทหน้ารายการสินค้าที่ต้องการลบ	ระบบทำการลบรายการสินค้าที่เลือกและแสดงข้อความ ("ข้อสินค้า") removed. "	ระบบทำการลบรายการสินค้าที่เลือกและแสดงข้อความ ("ข้อสินค้า" "removed.")	ผ่าน							
TC06	ผู้ใช้ทำการ Check out	ผู้ใช้คลิกปุ่ม PROCEED TO CHECK OUT	ไปหน้า Check Out	ไปหน้า Check Out	ผ่าน							

โครงการ :Katalon Shop												
รหัสสถานการณ์การทดสอบ : SC04	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสราวดัน แก้วปือก											
ฟังก์ชันที่ทดสอบ : CHECK OUT	วันที่เขียน : 10 ตุลาคม 2568											
เงื่อนไขก่อนทดสอบ : เพิ่มสินค้าลงในรายการรถเข็นสำเร็จ และกดปุ่ม PROCEED TO CHECK OUT												
เงื่อนไขหลังทดสอบ :												
ขั้นตอนการดำเนินการทดสอบ : ผู้ใช้คลิกลิงก์ Katalon Shop – Katalon Ecommerce												
<ul style="list-style-type: none"> - ผู้ใช้ทำการ Login เข้าสู่ระบบสำเร็จ - ผู้ใช้เลือกรายการ Shop - ผู้ใช้คลิกปุ่ม ADD TO CART ตรงสินค้าที่ต้องการเพิ่ม - ผู้ใช้เพิ่มสินค้าลงรายการรถเข็นสำเร็จ - ผู้ใช้เลือกรายการ CART - ผู้ใช้คลิกปุ่ม PROCEED TO CHECK OUT - ผู้ใช้คลิกปุ่ม PLACE ORDER 												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TC07	ผู้ใช้ยืนยันที่อยู่การจัดส่ง	ผู้ใช้คลิกปุ่ม PLACE ORDER	หน้าจอแสดงข้อมูลที่อยู่ และชื่อเจ้าของที่อยู่ที่จัดส่ง ถูกต้อง	หน้าจอแสดงข้อมูลที่อยู่ และชื่อเจ้าของที่อยู่ที่จัดส่ง ถูกต้อง	ผ่าน							
TC08	ผู้ใช้เลือกจัดส่งที่อื่น	ผู้ใช้ทำการคลิกถูกหน้าช่อง Ship to a different address? และคลิกปุ่ม PLACE ORDER	หน้าจอแสดงข้อมูลที่อยู่ การจัดส่งใหม่	หน้าจอแสดงข้อมูลที่อยู่การจัดส่งใหม่	ผ่าน							

โครงการ: Katalon Shop											
รหัสสถานการณ์การทดสอบ : SC05		ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปือก									
ฟังก์ชันที่ทดสอบ : Log Out		วันที่เขียน : 10 ตุลาคม 2568									
เงื่อนไขก่อนทดสอบ : ผู้ใช้ทำการ Login เข้าสู่ระบบสำเร็จ											
เงื่อนไขหลังทดสอบ :											
ขั้นตอนการดำเนินการทดสอบ : ขั้นตอนการดำเนินการทดสอบ											
:- ผู้ใช้คลิกลิงก์ Katalon Shop – Katalon Ecommerce											
- ผู้ใช้ทำการ Login เข้าสู่ระบบสำเร็จ											
- ผู้ใช้คลิกเลือกรายการ MY ACCOUNT											
- ผู้ใช้คลิก Log Out											
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ					
TC09	ผู้ใช้ทำการ Log Out	ผู้ใช้คลิกที่ Log Out	กลับไปหน้า Login	กลับไปหน้า Login	ผ่าน						

โครงการ CURA Healthcare Service											
รหัสสถานการณ์การทดสอบ : SC01		ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปือก									
ฟังก์ชันที่ทดสอบ : Login		วันที่เขียน : 10 ตุลาคม 2568									
เงื่อนไขก่อนทดสอบ :											
- ผู้ใช้คลิกลิงก์ https://katalon-demo-cura.herokuapp.com/appointment.php#summary											
เงื่อนไขหลังทดสอบ :											
ขั้นตอนการดำเนินการทดสอบ :											
- ผู้ใช้คลิกลิงก์ https://katalon-demo-cura.herokuapp.com/appointment.php#summary											
- ผู้ใช้เลือก Menu Login											
- ผู้ใช้ทำการกรอก Username และ Password											
- ผู้ใช้กดปุ่ม Login											
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ					
TC01	Login เข้าสู่ระบบสำเร็จ	Username : "John Doe" Password : "ThisIsNotAPassword"	ไปหน้า Make Appointment	ไปหน้า Make Appointment	ผ่าน						
TC02	Login เข้าสู่ระบบไม่สำเร็จ	Username : "John Doe" Password : "ThisIsNotAPass"	แจ้งเตือน "Password ไม่ถูกต้อง"	แจ้งเตือน "Login failed !Please ensure the username and password are valid"	ไม่ผ่าน						

โครงการ CURA Healthcare Service										
รหัสสถานการณ์การทดสอบ : SC02			ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก							
ฟังก์ชันที่ทดสอบ : Make Appointment			วันที่เขียน : 10 ตุลาคม 2568							
เงื่อนไขก่อนทดสอบ :										
<ul style="list-style-type: none"> - ผู้ใช้คลิกลิงค์:CURA Healthcare Service - ผู้ใช้ทำการล็อกอินสำเร็จ 										
เงื่อนไขหลังทดสอบ :										
<p>ขั้นตอนการดำเนินการทดสอบ :</p> <ul style="list-style-type: none"> - ผู้ใช้คลิกลิงค์ : CURA Healthcare Service - ผู้ใช้ทำการล็อกอินสำเร็จ - ผู้ใช้เลือกรายการ Make Appointment - ผู้ใช้เลือกรายการ Facility, Healthcare program, VisitDate - ผู้ใช้กดปุ่ม Book Appointment 										
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ				
TC03	ผู้ใช้ทำการจองสำเร็จ	Facility:"Tokyo CURA HealthCare Center" Healthcare program:"Medicare" VisitDate:8/10/2025 Comment:"Routine Checkup"	ไม่ปังหน้า Appoint Confirmation	ไม่ปังหน้า Appointment Confirmation	ผ่าน					
TC04	ผู้ใช้กรอกข้อมูลไม่ครบ	Facility:"Tokyo CURA HealthCare Center" Healthcare program:"Medicare" VisitDate:Null Comment:"Routine Checkup"	หน้าจอแสดงข้อความแจ้งเตือน"กรุณากรอกข้อมูลให้ครบ"	หน้าจอแสดงข้อความแจ้งเตือน"กรุณากรอกข้อมูลในช่องนี้"	ไม่ผ่าน					

โครงการ CURA Healthcare Service												
รหัสสถานการณ์การทดสอบ : SC03	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก											
ฟังก์ชันที่ทดสอบ : History	วันที่เขียน : 10 ตุลาคม 2568											
เงื่อนไขก่อนทดสอบ :												
<ul style="list-style-type: none"> - ผู้ใช้ทำการจองสำเร็จ 												
เงื่อนไขหลังทดสอบ :												
<p>ขั้นตอนการดำเนินการทดสอบ :</p> <ul style="list-style-type: none"> - ผู้ใช้คลิกลิงก์ https://katalon-demo-cura.herokuapp.com/appointment.php#summary - ผู้ใช้ทำการเลือกอ่อนสำเร็จ - ผู้ใช้ทำการจองสำเร็จ - ผู้ใช้กด Go to Homepage - ผู้ใช้กดเลือกปุ่มเมนู History 												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TC05	ผู้ใช้เลือกเมนู History	ผู้ใช้คลิกปุ่ม History	แสดงข้อมูลประวัติการนัด	แสดงข้อมูลประวัติการนัด	ผ่าน							

โครงการ CURA Healthcare Service												
รหัสสถานการณ์การทดสอบ : SC04	ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอก											
ฟังก์ชันที่ทดสอบ : Logout	วันที่เขียน : 10 ตุลาคม 2568											
เงื่อนไขก่อนทดสอบ :												
<ul style="list-style-type: none"> - ผู้ใช้ทำการ Login เข้าสู่ระบบสำเร็จ 												
เงื่อนไขหลังทดสอบ :												
<p>ขั้นตอนการดำเนินการทดสอบ :</p> <ul style="list-style-type: none"> - ผู้ใช้คลิกลิงก์ https://katalon-demo-cura.herokuapp.com/appointment.php#summary - ผู้ใช้ทำการ Login - ผู้ใช้กดเลือก Menu Logout 												
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ						
TC06	ผู้ใช้เลือก Menu Logout	ผู้ใช้กดและเลือก Logut	กลับไปหน้า Home page	กลับไปหน้า Home page	ผ่าน							

โครงการ CURA Healthcare Service											
รหัสสถานการณ์การทดสอบ : SC05		ผู้เขียนสถานการณ์การทดสอบ : นางสาวนิสารัตน์ แก้วปีอุก									
ฟังก์ชันที่ทดสอบ : Make Appointment		วันที่เขียน : 10 ตุลาคม 2568									
เงื่อนไขก่อนทดสอบ :											
<ul style="list-style-type: none"> - ผู้ใช้ทำการล็อกอินสำเร็จ 											
เงื่อนไขหลังทดสอบ :											
<p>ขั้นตอนการดำเนินการทดสอบ :</p> <ul style="list-style-type: none"> - ผู้ใช้คลิกลิงก์ https://katalon-demo-cura.herokuapp.com/appointment.php#summary - ผู้ใช้ทำการ Login สำเร็จ 											
รหัส	กรณีทดสอบ	ข้อมูลทดสอบ	ผลลัพธ์ที่คาดหวัง	ผลลัพธ์ที่เกิดขึ้นจริง	ผลการทดสอบ	หมายเหตุ					
TC07	ผู้ใช้กรอกข้อมูล ไม่ครบ	Faciliey : "Tokyo CURA Health Care Center" Healthcare Program : "Madicare" VisitDate : *ไม่เลือกวันที่ใส่comment "Routine checkup"	แสดงข้อมูลข้อความแจ้งเตือน "กรุณากรอกข้อมูลให้ครบ" แสดงข้อความ "กรุณากรอกข้อมูลในช่องนี้"		ผ่าน						

สมาชิกกลุ่ม

1. นางสาวนรมน พิลาภุล 6630200306
 2. นางสาวนิสรารัตน์ แก้วปีอก 6630200331
 3. นางสาวเสาวลักษณ์ บัญรอด 6630200667