

# Тестирање за прифатливост на BookShop веб-апликација со RobotFramework

ТЕСТИРАЊЕ НА CART МОДУЛ

СТАНИКЕВСКА ТАМАРА, 181081

## Содржина

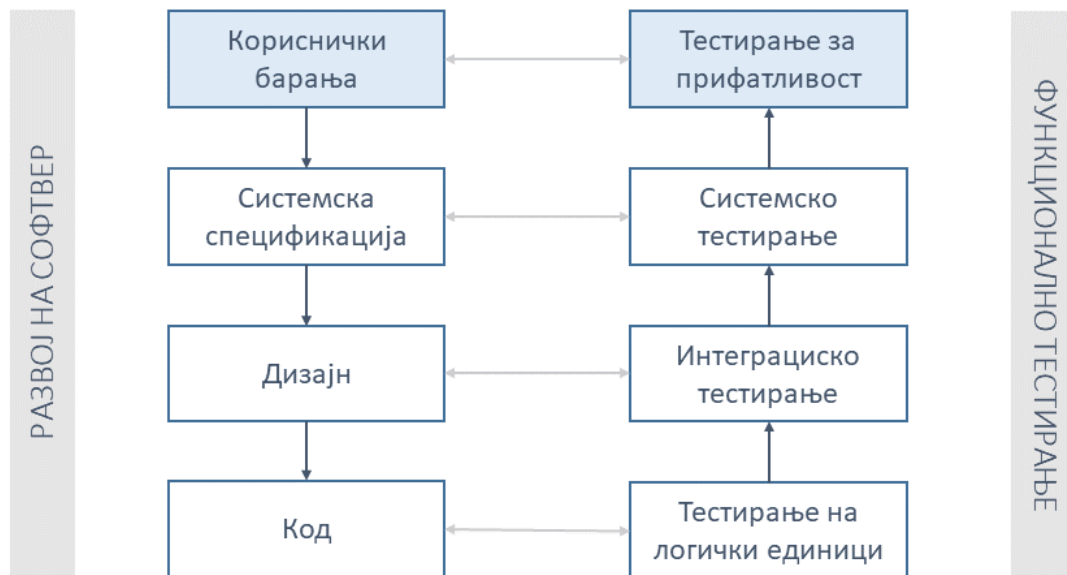
<b>Асептaнсе тестирање .....</b>	<b>3</b>
Тип на тестирање .....	3
Метод за тестирање .....	4
Техника/стратегија за тестирање .....	5
Тестирање базирано на краен корисник .....	5
Тестирање според промена на состојба .....	5
Тестирање на гранични вредности .....	5
Тестирање според табели на одлука .....	5
<b>Кориснички барања .....</b>	<b>6</b>
Корисничка кошничка .....	6
<b>Тестови .....</b>	<b>7</b>
Тестови базирани на краен корисник/системски тестови .....	7
Имплементација .....	9
Тестови за гранични вредности .....	11
Имплементација .....	11
Тестови за промена на состојба .....	12
Тестови базирани на табела на одлуки .....	14
Имплементација .....	14
<b>Извршување на тестовите .....</b>	<b>15</b>
<b>Заклучок .....</b>	<b>15</b>
<b>Користена литература .....</b>	<b>16</b>

## Листа на табели

Table 1: Тестови базирани на краен корисник.....	8
Table 2: DSL имплементација за end-user тестови.....	10
Table 3: Тестови за гранични вредности .....	11
Table 4: DSL имплементација за boundary тестови.....	11
Table 5: Тестови за промена на состојба .....	13
Table 6: Табела за одлука .....	14
Table 7: DSL имплементација за decision table тестови .....	14

## Acceptance тестирање

**Acceptance** или тестирање за прифатливост е последната фаза на функционално тестирање на некој софтверски продукт. Главната цел на овој метод е да се провери усогласеноста на системот со корисничките барања/приказна и да се оцени дали е прифатлив за испорака или не.



1 Фази на функционално тестирање

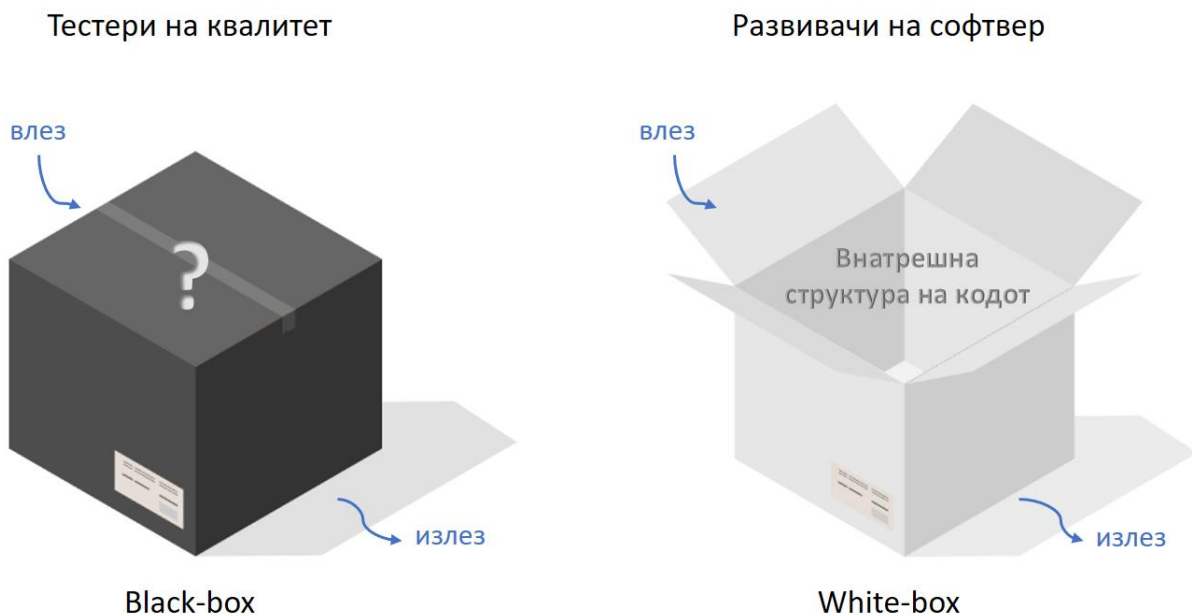
## Тип на тестирање

Има повеќе типови на тестирање за прифатливост, но голем дел се извршуваат од самите крајни корисници на системот. Тестирањето кое го вршат програмерите од тимот се нарекува **Алфа** тестирање.

## Метод за тестирање

Acceptance најчесто го користи **black-box** методот за тестирање кој уште и се нарекува тестирање базирано на однесување/спецификации/влез-излез. Овој методот се базира на евалуација на функционалноста на системот без да се интересира за интерната структура на кодот. Тестирањето се извршува според однесувањето на системот, а корисничките барања се самиот тест критериум. Секој тест се состои од некои влезни податоци (точни или грешни), а излезните податоци треба да одговараат на очекуваните. Поточно секој тест за прифатливост репрезентира некој очекуван резултат од системот.

Во зависност од потребите има случаи кога Alpha користи и white-box тестирање врз основа на структурата во кодот за отстранување на грешки.



2 Методи за тестирање (black-box наспроти white-box)

## Техника/стратегија за тестирање

За целосно тестирање на прифатливоста на системот потребно е да се користат неколку техники за опфаќање на сите тест случаи. Black-box методот има неколку типични техники за дизајн на тестови како што се: системски тестови базирани на краен корисник, тестирање според промена на состојба, тестирање на гранични вредности и тестирање според табели на одлука кои ќе се користат во овој документ и со нивна комбинација ќе се тестира корисничката кошничка.

### *Тестирање базирано на краен корисник*

- Системот што се тестира може да има многу компоненти кои кога се споени заедно го постигнуваат корисничкото сценарио. Со оваа техника се тестираат одделните компоненти како и комплетни сценарија базирани според крајниот корисник. Овие тестови уште се нарекуваат и системски тестови.

### *Тестирање според промена на состојба*

- Тестирање на транзиција на состојби е техника која се користи за тестирање на различните состојби на системот што се тестира. Состојбата на системот се менува во зависност од условите или настаните.
- Се креира дијаграм за систематска транзиција на состојби кој дава јасен приказ на промените на состојбата.

### *Тестирање на гранични вредности*

- Оваа техника се фокусира на формирање тестови за граничните вредности на некои функционалности во апликацијата каде што се менува однесувањето на системот.

### *Тестирање според табели на одлука*

- Од условите на системот и функционалностите што треба да ги изврши системот се креира табела на одлуки според сите можни сценарија. Секоја комбинација преставува посебен тест случај.

## Кориснички барања

За тестирање за прифатливост најважно е да се провери дали корисничките барања за системот се задоволени. Во продолжение е дадена листа од функционалните барања за Cart модулот кој се тестира во овој документ.

### Корисничка кошничка

- 1. Системот треба да овозможи додавање на книги во корисничка кошничка.*
  - Додавање на книги од почетната страна.*
  - Додавање на книги од страната за детален приказ на книгата.*
- 2. Системот треба да го прикажува бројот на книги кои што се во кошничка на почетната страна.*
- 3. Системот треба да овозможи преглед на книгите кои се додадени во кошничката.*
- 4. Системот треба да овозможи преглед на количината на секоја книга одделно во кошничката.*
- 5. Системот треба да овозможи бришење на книги од кошничката.*
- 6. Системот треба да има копче за промена на количината на книгите во кошничката.*
- 7. Системот треба да ја прикажува вкупната сума за секоја книга во кошничката.*
- 8. Системот треба да има копче во корисничката кошничка за продолжување со купувањето.*
- 9. Системот треба да овозможи најава на корисникот за процесирање на нарачката.*
- 10. Системот треба да го пренасочи корисникот на почетната страна по најавата.*
- 11. Системот треба да има копче за процесирање на нарачката.*
- 12. Системот треба да го пренасочи корисникот на страната со сите книги при клик на копчето за процесирање кога е празна кошничката.*
- 13. Системот треба да го пренасочи корисникот на страната за внес на лични информации при клик на копчето за процесирање кога е полна кошничката.*

## Тестови

За тестирање за прифатливост на Cart модулот се користи алатката **Robot Framework**. Водејќи се според корисничките барања за системот и користејќи го претходно дефинираниот метод и стратегија се креирани тест случаите. За полесна имплементација на тестовите и нивно разбирање по пишувањето на кодот, се креирани табели кои соодветствуваат на Domain Specific Language на RobotFramework, табела на одлуки и дијаграм за промена на состојби.

### Тестови базирани на краен корисник/системски тестови

Во следните две табели (табела 1 и табела 2) се дадени тестовите креирани според крајните корисници. Имплементацијата на овие тестови е дадена во датотека end\_user\_tests.robot.

#	Тест случај за:	Акции:	Резултат:	Статус:
1	Постоење на копче за корисничка кошничка на почетната страница.	Корисникот ја отвара почетната страница.	Копчето е поставено на навигациското мени.	P
2	Успешно додавање книга од почетната страна.	Корисникот клика на "Add to cart".	Книгата е излистана во кошничката.	P
		Корисникот клика на "Cart".		
3	Успешно додавање на книга од страната за детален приказ на книгата	Корисникот клика на книгата.	Книгата е излистана во кошничката.	P
		Корисникот клика на "Add to cart".		
		Корисникот клика на "Cart".		
4	Приказ на бројот на книги кои што се во кошничка на почетната страна.	Корисникот клика на "Add to cart".	Бројот на книги во листата на "Cart" е зголемен за еден.	P



#	Тест случај за:	Акција	Резултат	Статус
5	Приказ на количината на една книга во корисничка кошничка.	Корисникот клика на "Add to cart".	Книгата и нејзината количина е излистана во кошничката.	P
		Корисникот клика на "Cart".		
6	Приказ на вкупната сума на книга во корисничката кошничка.	Корисникот клика на "Add to cart".	Книгата и нејзината вкупна сума е прикажана во кошничката.	P
		Корисникот клика на "Add to cart".		
		Корисникот клика на "Cart".		
7	Бришење на книга од корисничката кошничка.	Корисникот клика на "Add to cart".	Кошничката е празна.	P
		Корисникот клика на "Cart".		
		Корисникот клика на копчето за бришење на книга.		
8	Зголемување на количината на книга во корисничка кошничка.	Корисникот клика на "Add to cart".	Количината на книгата во кошничката е зголемена за еден.	P
		Корисникот клика на "Cart".		
		Корисникот клика на плусот.		
9	Намалување на количината на книга во корисничка кошничка.	Корисникот клика на "Add to cart".	Количината на книгата во кошничката е намалена за еден.	P
		Корисникот клика на "Add to cart".		
		Корисникот клика на "Cart".		
		Корисникот клика на минусот.		

Table 1 Тестови базирани на краен корисник

\*\* Во тестовите се подразбира дека со стартување на апликацијата корисникот се наоѓа на почетната страница.

Имплементација

#	Test Case	Keyword	Argument1	Argument2
1	Home Page Should Contain Cart Button	Open App		
		Has Cart Button	True	
2	Add To Cart From Home Page	Add Book To Cart	Book1	
		Open Shopping Cart		
		Has Book In Cart	True	
3	Add To Cart From Details Page	Open Details Page For Book	Book1	
		Add Book To Cart		
		Open Shopping Cart		
		Has Book In Cart	True	
4	Show Cart Quantity On Home Page	Add Book To Cart	Book1	
		Get Total In Cart		
		Should Be Equal	Total In Cart	Expected Quantity
5	Show Book Quantity In Cart	Add Book To Cart	Book1	
		Open Shopping Cart		
		Get Book Quantity In Cart		
		Should Be Equal	Book Quantity In Cart	Expected Quantity
6	Show Book Sum In Cart	Add Book To Cart	Book1	x2
		Get Book Price	Book1	
		Open Shopping Cart		
		Should Be Equal	Book Price	Cart Sum

#	Test Case	Keyword	Argument1	Argument2
7	Delete Book From Cart	Add Book To Cart	Book1	
		Open Shopping Cart		
		Delete Book		
		Cart Is Empty	True	
8	Increase Book Quantity In Cart	Add Book To Cart	Book1	
		Open Shopping Cart		
		Click Plus		
		Get Book Quantity In Cart		
		Should Be Equal	Book Quantity In Cart	Expected Quantity
9	Decrease Book Quantity In Cart	Add Book To Cart	Book1	x2
		Open Shopping Cart		
		Click Minus		
		Get Book Quantity In Cart		
		Should Be Equal	Book Quantity In Cart	Expected Quantity

Table 2 DSL имплементација за end-user тестови

## Тестови за гранични вредности

Во следните две табели (табела 3 и табела 4) е тестот за гранички вредности во корисничката кошничка. Имплементацијата на овој тест е дадена во датотека boundary\_tests.robot.

#	Тест случај	Акција	Резултат	Статус
1	Намалување на количината на книга во корисничка кошничка кога има само една книга	Корисникот клика на "Add to cart".	Количината на книгата останува еден.	P
		Корисникот клика на "Cart".		
		Корисникот клика на минусот.		

Table 3 Тестови за гранични вредности

## Имплементација

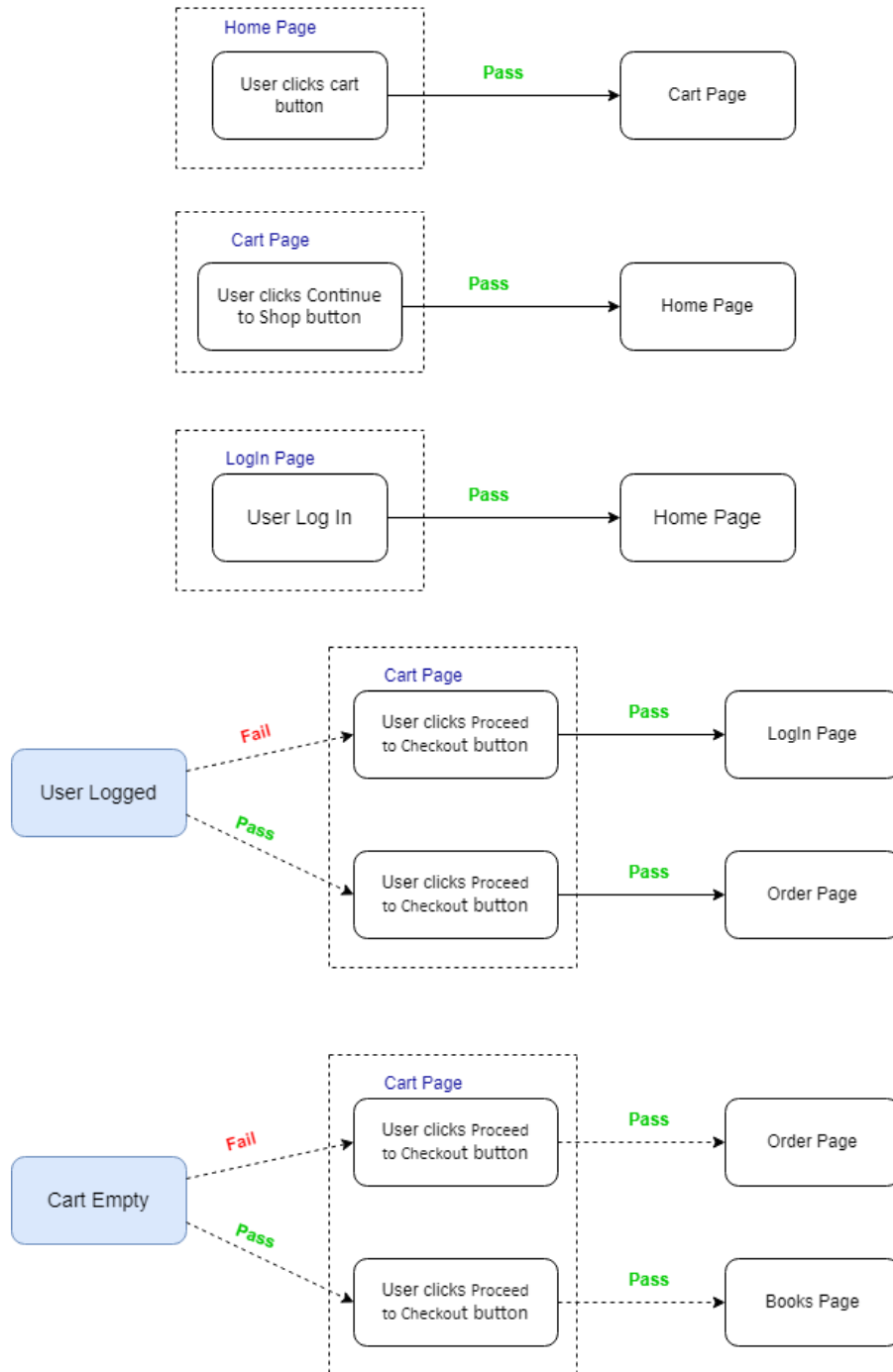
#	Test Case	Keyword	Argument1	Argument2
1	Cart Quantity Boundary	Add Book To Cart	Book1	
		Open Shopping Cart		
		Click Minus		
		Get Book Quantity In Cart		
		Should Be Equal	Book Quantity	1

Table 4 DSL имплементација за boundary тестови

## Тестови за промена на состојба

На дијаграмот во продолжение (слика 3) се прикажани промените на состојба на модулот кој што се тестира од веб апликацијата. Според овој дијаграм се креирани тестовите во табела 5 подоле.

Имплементацијата на овие тестови е дадена во датотека `state_transition_tests.robot`.



3 Дијаграм за промена на состојби

#	Тест случај за:	Акција	Резултат	Статус
1	Приказ на корисничка кошничка.	Корисникот клика на "Cart".	Корисникот е пренасочен на страната за корисничка кошничка.	P
2	Продолжување со купување.	Корисникот клика на "Cart". Корисникот клика на "Continue to Shop".	Корисникот е пренасочен на почетната страна.	P
3	Продолжување со наплата кога корисникот не е најавен.	Корисникот клика на "Cart". Корисникот клика на "Proceed to Checkout".	Корисникот е пренасочен на страната за најава.	P
4	Успешна најава на корисникот.	Корисникот се наоѓа на страната за најава. Корисникот внесува важечки кредитни податоци. Корисникот клика на "Submit".	Корисникот е пренасочен на почетната страна.	P
5	Продолжување со наплата кога корисникот е најавен.	Корисникот се најавува. Корисникот клика на "Add to cart". Корисникот клика на "Cart". Корисникот клика на "Proceed to Checkout".	Корисникот е пренасочен на страната за внес на лични информации.	P
6	Продолжување со наплата кога кошничката е празна.	Корисникот се најавува. Корисникот клика на "Cart". Корисникот клика на "Proceed to Checkout".	Корисникот е пренасочен на страната со книги.	P
7	Продолжување со наплата кога кошничката е полна.	Корисникот се најавува. Корисникот клика на "Add to cart". Корисникот клика на "Cart". Корисникот клика на "Proceed to Checkout".	Корисникот е пренасочен на страната за внес на лични информации.	P

Table 5 Тестови за промена на состојба

## Тестови базирани на табела на одлуки

Тестовите базирани на одлука се дадени во табела 6 , како и табела 7 која соодветствува на Robot DSL за секој тест случај. Имплементацијата на овие тестови се наоѓа во decision\_tests.robot

<b>Тест случаи за продолжување со наплата</b>				
Услови				
Корисникот е најавен	F	F	T	T
Кошничката е полна	F	T	F	T
Акции	T4	T3	T2	T1
Корисникот е пренасочен на страната за најава	x	x		
Корисникот е пренасочен на страната со книги			x	
Корисникот е пренасочен на страната за внес на лични информации				x

Table 6 Табела за одлука

## Имплементација

#Test Case	Keyword	Action	Argument
1	User should be logged Cart should not be empty	Proceed to Checkout	
		Page is	Order
2	User should be logged Cart should be empty	Proceed to Checkout	
		Page is	Books
3	User should not be logged Cart should not be empty	Proceed to Checkout	
		Page is	Login
4	User should not be logged Cart should be empty	Proceed to Checkout	
		Page is	Login

Table 7 DSL имплементација за decision table тестови

## Извршување на тестовите

Тестовите автоматски се извршуваат со скриптата `run_tests.py`. Истото може да се направи и од терминал со следната командна линија: `'robot -d robot_tests/test_results robot_tests/*.robot '`

Извештајот за сите тестови е прикажан во `log.html`, `report.html` и `output.html` кои се наоѓаат во папката `test_results`.

## Заклучок

Сите тест случаи наведени погоре се успешно извршени со помош на алатката Robot Framework. Од тука може да се заклучи дека модулот **Cart** ги задоволува корисничките барања и е прифатлив за употреба.



## Користена литература

- [1] <https://www.softwaretestinghelp.com/functional-testing-vs-non-functional-testing/>
- [2] <https://www.geeksforgeeks.org/acceptance-testing-software-testing/>
- [3] <https://www.digite.com/agile/acceptance-testing/>
- [4] [https://en.wikipedia.org/wiki/Black-box\\_testing](https://en.wikipedia.org/wiki/Black-box_testing)
- [5] <https://www.softwaretestinghelp.com/black-box-testing/>
- [6] <https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>
- [7] <http://robotframework.org/robotframework/2.7/libraries/BuiltIn.html>
- [8] <https://robotframework.org/robotframework/latest/libraries/String.html>
- [9] <https://robotframework.org/robotframework/latest/libraries/Collections.html>