



Miroslav Dionisiev and Pavel Sarlov

Test Cases

Faculty of mathematics and informatics

Course: Software quality assurance

Students: Miroslav Dionisiev, FN: 62390

Pavel Sarlov, FN: 62393

Table of contents

1.	Test Home Page	6
1.1	Test Information	6
1.1.1	Test type	6
1.1.2	System Under Test	6
1.1.3	Test Personnel	6
1.2	Test Summary	6
1.2.1	Results	6
1.3	Test Cases	7
1.3.1	Test “РАЗПИСАНИЕ” Section	7
1.3.2	Test “БИЛЕТЫ” Section	8
1.3.3	Test “АКТУАЛНО” Section	9
1.3.4	Test “LIVE” Section	10
1.3.5	Test “РАДАР” Section	11
1.3.6	Test “СИГНАЛИ” Section	12
1.4	Traceability matrix	13
2.	Test Train Timetable	14
2.1	Test Information	14
2.1.1	Test type	14
2.1.2	System Under Test	14
2.1.3	Test Personnel	14
2.2	Test Summary	14
2.2.1	Results	14
2.3	Test Cases	15
2.3.1	Test timetable departure and arrival stations only	15
2.3.2	Test timetable departure and arrival stations with departure date	16
2.3.3	Test timetable departure and arrival stations with return date	17
2.3.4	Test timetable departure and arrival stations with past date	18
2.3.5	Test timetable departure and arrival stations with middle station (valid)	19
2.3.6	Test timetable departure and arrival stations with middle station (invalid)	20

2.3.7 Test timetable departure and arrival stations with train type	21
2.3.8 Test timetable with train number (valid)	22
2.3.9 Test timetable with train number (invalid)	23
2.4 Traceability matrix.....	24
3. Test Tickets Registration	25
3.1 Test Information.....	25
3.1.1 Test type.....	25
3.1.2 System Under Test	25
3.1.3 Test Personnel	25
3.2 Test Summary	25
3.2.1 Results	25
3.3 Test Cases.....	26
3.3.1 Test registration form with valid data	26
3.3.2 Test registration form with invalid first name.....	27
3.3.3 Test registration form with invalid last name	28
3.3.4 Test registration form with invalid email.....	29
3.3.5 Test registration form with invalid password	30
3.3.6 Test registration form with invalid confirmation password	31
3.3.7 Test registration form with invalid phone	32
3.4 Traceability matrix.....	33
4. Test Tickets Login	34
4.1 Test Information.....	34
4.1.1 Test type.....	34
4.1.2 System Under Test	34
4.1.3 Test Personnel	34
4.2 Test Summary	34
4.2.1 Results	34
4.3 Test Cases.....	35
4.3.1 Test login form with valid data	35
4.3.2 Test login form with invalid email	36
4.3.3 Test login form with invalid password	37

4.4 Traceability matrix.....	38
5. Test Tickets Reservation	39
5.1 Test Information.....	39
5.1.1 Test type.....	39
5.1.2 System Under Test	39
5.1.3 Test Personnel	39
5.2 Test Summary	39
5.2.1 Results	39
5.3 Test Cases.....	40
5.3.1 Test searching departure and arrival stations one-way ticket.....	40
5.3.2 Test searching departure and arrival stations two-way ticket.....	41
5.3.3 Test searching without departure and arrival stations	42
5.3.4 Test ticket reservation for one person.....	43
5.3.5 Test ticket reservation for two people	45
5.3.6 Test session time expiration.....	47
5.4 Traceability matrix.....	48
6. Test Live	49
6.1 Test Information.....	49
6.1.1 Test type.....	49
6.1.2 System Under Test	49
6.1.3 Test Personnel	49
6.2 Test Summary	49
6.2.1 Results	49
6.3 Test Cases.....	50
6.3.1 Test live tracer with valid departure station.....	50
6.3.2 Test live tracer with valid arrival station	51
6.3.3 Test live tracer with invalid station	52
6.4 Traceability matrix.....	53
7. Test Radar	54
7.1 Test Information.....	54
7.1.1 Test type.....	54

7.1.2 System Under Test	54
7.1.3 Test Personnel	54
7.2 Test Summary	54
7.2.1 Results	54
7.3 Test Cases.....	55
7.3.1 Test radar with active train and its route	55
7.3.2 Test radar with train number	56
7.4 Traceability matrix.....	57

1. Test Home Page

1.1 Test Information

1.1.1 Test type

- ☒ Functional Test
☐ Performance Test

1.1.2 System Under Test

System name: BDZ Passenger Services

Version: 1.0

Short description of the system:

Online system which provides railway travel services like timetables, ticket reservation, live train tracing, news and information, etc.

1.1.3 Test Personnel

Name: Miroslav Dionisiev Date: 07-01-2022 Time/h: 0.5

Name: Pavel Sarlov Date: 07-01-2022 Time/h: 0.5

1.2 Test Summary

1.2.1 Results

Total number of test cases:	6
Total number of test cases passed:	6
Total number of test cases failed:	0
Total number of bugs found:	0

1.3 Test Cases

1.3.1 Test “РАЗПИСАНИЕ” Section

Special Instructions

NONE

Test Case ID	TC_FUNCT_01_01	
Description	Tests home page section “РАЗПИСАНИЕ”	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the home page.	The home page is loaded.
2	Verify the “РАЗПИСАНИЕ” button is present.	The button is present.
3	Click the “РАЗПИСАНИЕ” button.	The “РАЗПИСАНИЕ” page is loaded.
4	Verify the correct page is loaded.	The correct page is loaded.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the “РАЗПИСАНИЕ” button is present
- *assert* is used to verify the “РАЗПИСАНИЕ” page is loaded.

1.3.2 Test “БИЛЕТЫ” Section

Special Instructions

NONE

Test Case ID	TC_FUNCT_01_02	
Description	Tests home page section “БИЛЕТЫ”	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the home page.	The home page is loaded.
2	Verify the “БИЛЕТЫ” button is present.	The button is present.
3	Click the “БИЛЕТЫ” button.	The “БИЛЕТЫ” page is loaded.
4	Verify the correct page is loaded.	The correct page is loaded.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the “БИЛЕТЫ” button is present
- *assert* is used to verify the “БИЛЕТЫ” page is loaded.
- Selenium IDE doesn't provide specific command to switch between tabs, so we conduct no further verifications.

1.3.3 Test “АКТУАЛНО” Section

Special Instructions

NONE

Test Case ID	TC_FUNCT_01_03	
Description	Tests home page section “АКТУАЛНО”	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the home page.	The home page is loaded.
2	Verify the “АКТУАЛНО” button is present.	The button is present.
3	Click the “АКТУАЛНО” button.	The “АКТУАЛНО” page is loaded.
4	Verify the correct page is loaded.	The correct page is loaded.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the “АКТУАЛНО” button is present
- *assert* is used to the verify the “АКТУАЛНО” page is loaded.

1.3.4 Test “LIVE” Section

Special Instructions

NONE

Test Case ID	TC_FUNCT_01_04	
Description	Tests the homepage section button “LIVE”	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the homepage.	The homepage is loaded.
2	Assert that there is a “LIVE” section button.	The “LIVE” section button is present and visible.
3	Click the “LIVE” section button.	Redirected to “LIVE” page.
4	Assert that the current URL is correct.	The URL is correct.
5	Assert the page is displayed correctly.	The page is displayed correctly.
Test verdict	Passed	

Comments:

- The URL is stored in a variable using the *execute script* command and some javascript code.
- The assertions are made using the *assert* command and its extensions.

1.3.5 Test “РАДАР” Section

Special Instructions

NONE

Test Case ID	TC_FUNCT_01_05	
Description	Tests the homepage section button “РАДАР”	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the homepage.	The homepage is loaded.
2	Assert that there is a “РАДАР” section button.	The “РАДАР” section button is present and visible.
3	Click the “РАДАР” section button.	Redirected to “РАДАР” page.
4	Assert that the current URL is correct.	The URL is correct.
5	Assert the page is displayed correctly.	The page is displayed correctly.
Test verdict	Passed	

Comments:

- The URL is stored in a variable using the *execute script* command and some javascript code.
- The assertions are made using the *assert* command and its extensions.

1.3.6 Test “СИГНАЛИ” Section

Special Instructions

NONE

Test Case ID	TC_FUNCT_01_05	
Description	Tests the homepage section button “СИГНАЛИ”	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the homepage.	The homepage is loaded.
2	Assert that there is a “СИГНАЛИ” section button.	The “СИГНАЛИ” section button is present and visible.
3	Click the “СИГНАЛИ” section button.	Redirected to “СИГНАЛИ” page.
4	Assert that the current URL is correct.	The URL is correct.
5	Assert the page is displayed correctly.	The page is displayed correctly.
Test verdict	Passed	

Comments:

- The URL is stored in a variable using the *execute script* command and some javascript code.
- The assertions are made using the *assert* command and its extensions.

1.4 Traceability matrix

Test Case ID	Bug Description	Note
--------------	-----------------	------

2. Test Train Timetable

2.1 Test Information

2.1.1 Test type

- ☒ Functional Test
☐ Performance Test

2.1.2 System Under Test

System name: BDZ Passenger Services

Version: 1.0

Short description of the system:

Online system which provides railway travel services like timetables, ticket reservation, live train tracing, news and information, etc.

2.1.3 Test Personnel

Name: Miroslav Dionisiev Date: 07-01-2022 Time/h: 0.5

Name: Pavel Sarlov Date: 07-01-2022 Time/h: 0.5

2.2 Test Summary

2.2.1 Results

Total number of test cases:	7
Total number of test cases passed:	6
Total number of test cases failed:	1
Total number of bugs found:	1

2.3 Test Cases

2.3.1 Test timetable departure and arrival stations only

Special Instructions

NONE

Test Case ID	TC_FUNCT_02_01	
Description	Test timetable departure and arrival stations only	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the "РАЗПИСАНИЕ" page.	The "РАЗПИСАНИЕ" page is loaded.
2	Type departure station name.	The station name is found in the dropdown list of stations.
3	Type arrival station name.	The station name is found in the dropdown list of stations.
4	Verify there is a "Наперед" button.	The button is present
5	Click the "Наперед" button.	The form is submitted.
6	Verify the correct page is loaded.	The page is displayed correctly
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the "Наперед" button is present
- *store text* is used to store what is the track of the trains in the loaded page
- *execute script* is used to edit the stored text to only what I need
- *assert* is used to verify the current track from the page is the same as the one expected

2.3.2 Test timetable departure and arrival stations with departure date

Special Instructions

NONE

Test Case ID	TC_FUNCT_02_02	
Description	Tests the timetable with departure date	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the timetable page.	The timetable page is loaded.
2	Assert there is a timetable search form.	The timetable search form is present.
3	Enter a departure station.	The departure station is entered.
4	Enter an arrival station.	The arrival station is entered.
5	Enter a valid departure date.	The departure date is entered.
6	Click the submit button.	The form is submitted.
7	Assert the displayed result is correct.	The displayed result is correct.
Test verdict	Passed	

Comments:

- The values of the stations and the date are stored in variables, using the *store value* command, which are later used to assert the result is correct.
- The assertions are made using the *assert* command and its extensions.

2.3.3 Test timetable departure and arrival stations with arrival date

Special Instructions

NONE

Test Case ID	TC_FUNCT_02_03	
Description	Test timetable departure and arrival stations with arrival date	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the "РАЗПИСАНИЕ" page.	The "РАЗПИСАНИЕ" page is loaded.
2	Type departure station name.	The station name is found in the dropdown list of stations.
3	Type arrival station name.	The station name is found in the dropdown list of stations.
4	Pick the "Пристига след" option in the dropdown menu	The label's content is changed to "Пристига след".
5	Verify that the "arrival" option is selected	The label holds correct value.
6	Pick arrival time from the dropdown menu	The label's content is changed.
7	Verify the set hour	The label's content corresponds to the expected value.
8	Verify there is a "Напред" button.	The button is present.
9	Click the "Напред" button.	A page with listed trains is loaded.
10	Verify the correct page is loaded	The page is displayed correctly.

Comments:

- *assert selected value* is used to verify correct values are selected from the drop down menus,
- *assert element present* is used to assert whether the "Напред" button is present
- *store text* is used to store what is the track of the trains in the loaded page
- *execute script* is used to edit the stored text to only what I need
- *assert* to verify the current track from the page is the same as the one expected.

2.3.4 Test timetable departure and arrival stations with past date

Special Instructions

NONE

Test Case ID	TC_FUNCT_02_04	
Description	Tests the timetable with past date	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the timetable page.	The timetable page is loaded.
2	Assert there is a timetable search form.	The timetable search form is present.
3	Enter a departure station.	The departure station is entered.
4	Enter an arrival station.	The arrival station is entered.
5	Enter a valid past departure date.	The departure date is entered.
6	Click the submit button.	The form is submitted.
7	Assert the displayed result is correct.	The displayed result is correct.
8	Assert there is a message for wrong criteria.	The message is displayed correctly.
Test verdict	Passed	

Comments:

- The values of the stations and the date are stored in variables, using the *store value* command, which are later used to assert the result is correct.
- The assertions are made using the *assert* command and its extensions.

2.3.5 Test timetable departure and arrival stations with middle station (valid)

Special Instructions

NONE

Test Case ID	TC_FUNCT_02_05	
Description	Test timetable departure and arrival stations with middle station (valid)	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the "РАЗПИСАНИЕ" page.	The "РАЗПИСАНИЕ" page is loaded.
2	Type departure station name.	The station name is found in the dropdown list of stations.
3	Type arrival station name.	The station name is found in the dropdown list of stations.
4	Type middle station name.	The station name is found in the dropdown list of stations.
5	Verify there is a "Наперед" button.	The button is present.
6	Click the "Наперед" button.	A page with listed trains is loaded.
7	Verify the correct page is loaded	The page is displayed correctly.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the "Наперед" button is present
- *store text* is used to store what is the track of the trains in the loaded page
- *execute script* is used to edit the stored text to only what I need
- *assert* is used to verify the current track from the page is the same as the one expected.

2.3.6 Test timetable departure and arrival stations with middle station (invalid)

Special Instructions

NONE

Test Case ID	TC_FUNCT_02_06	
Description	Tests the timetable with invalid middle station	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the timetable page.	The timetable page is loaded.
2	Assert there is a timetable search form.	The timetable search form is present.
3	Assert there is a form for options.	The form for options is present.
3	Enter a departure station.	The departure station is entered.
4	Enter an arrival station.	The arrival station is entered.
5	Enter an invalid middle station.	The middle station is entered.
6	Click the submit button.	The form is submitted.
7	Assert the displayed result is correct.	The displayed result is correct.
8	Assert there is a message for wrong criteria.	The message is displayed correctly.
Test verdict	Failed	

Comments:

- The values of the stations and the date are stored in variables, using the *store value* command, which are later used to assert the result is correct.
- The assertions are made using the *assert* command and its extensions.
- The server throws an error 500 which is unusual.

2.3.7 Test timetable departure and arrival stations with train type

Special Instructions

NONE

Test Case ID	TC_FUNCT_02_07	
Description	Test timetable departure and arrival stations with train type	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the "РАЗПИСАНИЕ" page.	The "РАЗПИСАНИЕ" page is loaded.
2	Type departure station name.	The station name is found dropdown list of stations.
3	Type arrival station name.	The station name is found dropdown list of stations.
4	Pick a train type from the checkbox list	A box is checked
5	Verify there is a "Наперед" button.	The button is present.
6	Click the "Наперед" button.	A page with listed trains is loaded.
7	Verify the correct page is loaded	The page is displayed correctly.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the "Наперед" button is present
- *store text* is used to store what is the track of the trains in the loaded page and whether the train type is correct
- *execute script* is used to edit the stored text to only what I need
- *assert* is used to verify the current track from the page is the same as the one expected.

2.3.8 Test timetable with train number (valid)

Special Instructions

NONE

Test Case ID	TC_FUNCT_02_08	
Description	Tests the timetable with valid train number	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the timetable page.	The timetable page is loaded.
2	Assert there is a timetable search form.	The timetable search form is present.
3	Assert there is a form for options.	The form for options is present.
5	Enter a valid train number.	The train number is entered.
6	Click the submit button.	The form is submitted.
7	Assert the displayed result is correct.	The displayed result is correct.
8	Assert the timetable is displayed.	The timetable is displayed.
Test verdict	Passed	

Comments:

- The values of the stations and the date are stored in variables, using the *store value* command, which are later used to assert the result is correct.
- The assertions are made using the *assert* command and its extensions.

2.3.9 Test timetable with train number (invalid)

Special Instructions

NONE

Test Case ID	TC_FUNCT_02_09	
Description	Tests the timetable with invalid train number	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions		
Test Step ID	Test Step Description	Expected Result
1	Open the "РАЗПИСАНИЕ" page.	The "РАЗПИСАНИЕ" page is loaded.
2	Type incorrect train number.	
3	Verify the presence of submit button.	The button is present.
4	Click the submit button.	The form is submitted.
5	Verify that the correct page is loaded.	The correct page is loaded.
Test verdict	Failed	

Comments:

- The assertions are made using the *assert element present* command.

2.4 Traceability matrix

Test Case ID	Bug Description	Note
TC_FUNCT_02_06	The server returns error 500 which is not handled.	The error should be handled and an appropriate message should be displayed.
TC_FUNCT_02_09	The server returns error 404 which is not handled.	The error should be handled and an appropriate message should be displayed.

3. Test Tickets Registration

3.1 Test Information

3.1.1 Test type

- ☒ Functional Test
☐ Performance Test

3.1.2 System Under Test

System name: BDZ Passenger Services

Version: 1.0

Short description of the system:

Online system which provides railway travel services like timetables, ticket reservation, live train tracing, news and information, etc.

3.1.3 Test Personnel

Name: Miroslav Dionisiev Date: 07-01-2022 Time/h: 0.5

Name: Pavel Sarlov Date: 07-01-2022 Time/h: 0.5

3.2 Test Summary

3.2.1 Results

Total number of test cases:	6
Total number of test cases passed:	5
Total number of test cases failed:	1
Total number of bugs found:	1

3.3 Test Cases

3.3.1 Test registration form with valid data

Special Instructions

NONE

Test Case ID	TC_FUNCT_03_01	
Description	Tests the registration of the tickets page	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the tickets page.	The tickets page is loaded.
2	Click the registration link.	Redirected to registration form.
3	Assert the registration form is displayed correctly.	The registration form is displayed correctly.
4	Enter a valid email.	The email is entered.
5	Enter a valid first name.	The first name is entered.
6	Enter a valid last name.	The last name is entered.
7	Enter a valid password.	The password is entered and masked.
8	Enter a valid confirmation password.	The confirmation password is entered and masked.
9	Enter a valid phone number.	The phone number is entered.
10	Check the terms of service prompt.	The prompt is checked.
11	Click the submit button.	The form is submitted.
12	Assert the registration was successful.	The registration was successful.
Test verdict	Passed	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed.
- The assertions are made using the *assert* command and its extensions.
- Command *execute script* is used to run a simple JS for a random string.

3.3.2 Test registration form with invalid first name

Special Instructions

NONE

Test Case ID	TC_FUNCT_03_02	
Description	Test registration form with invalid first name	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the "БИЛЕТИ" page.	The "БИЛЕТИ" page is loaded.
2	Verify the presence of registration button.	The registration button is present.
3	Click the registration button.	Open the registration page.
4	Fill the registration form.	The registration form has incorrect first name.
5	Verify the presence of checkbox for confirmation the terms of use.	The checkbox is present.
6	Check the box.	
7	Verify there is a registration button.	The registration button is present.
8	Register a user	Click the register button.
9	Check if there is a text box with a warning.	The user is warned about the incorrect first name.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the Registration button is present
- *assert check* is used to verify the checkbox is checked
- *store text* is used to get the content of the warning
- *execute script* is used to generate random names and emails
- *assert* is used to compare the one received and the expected one.

3.3.3 Test registration form with invalid last name

Special Instructions

NONE

Test Case ID	TC_FUNCT_03_03	
Description	Tests the registration of the tickets page with invalid last name	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the tickets page.	The tickets page is loaded.
2	Click the registration link.	Redirected to registration form.
3	Assert the registration form is displayed correctly.	The registration form is displayed correctly.
4	Enter a valid email.	The email is entered.
5	Enter a valid first name.	The first name is entered.
6	Enter an invalid last name.	The last name is entered.
7	Enter a valid password.	The password is entered and masked.
8	Enter a valid confirmation password.	The confirmation password is entered and masked.
9	Enter a valid phone number.	The phone number is entered.
10	Check the terms of service prompt.	The prompt is checked.
11	Click the submit button.	The form is submitted.
12	Assert the registration was unsuccessful and an appropriate alert message is displayed.	The registration failed and an alert message is displayed.
Test verdict	Passed	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed.
- The assertions are made using the *assert* command and its extensions.
- Command *execute script* is used to run a simple JS for a random string.

3.3.4 Test registration form with invalid email

Special Instructions

NONE

Test Case ID	TC_FUNCT_03_04	
Description	Test registration form with invalid email	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the "БИЛЕТИ" page.	The "БИЛЕТИ" page is loaded.
2	Verify the presence of registration button.	The registration button is present.
3	Click the registration button.	Open the registration page.
4	Fill the registration form.	The registration form has incorrect email.
5	Verify the presence of checkbox for confirmation the terms of use.	The checkbox is present.
6	Check the box.	
7	Verify there is a registration button.	The registration button is present.
8	Register a user	Click the register button.
9	Check if there is a text box with a warning.	The user is warned about the incorrect email.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the Registration button is present
- *assert check* is used to verify the checkbox is checked
- *store text* is used to get the content of the warning
- *execute script* is used to generate random names and emails
- *assert* is used to compare the one received and the expected one.

3.3.5 Test registration form with invalid password

Special Instructions

NONE

Test Case ID	TC_FUNCT_03_05	
Description	Tests the registration of the tickets page with invalid password	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the tickets page.	The tickets page is loaded.
2	Click the registration link.	Redirected to registration form.
3	Assert the registration form is displayed correctly.	The registration form is displayed correctly.
4	Enter a valid email.	The email is entered.
5	Enter a valid first name.	The first name is entered.
6	Enter a valid last name.	The last name is entered.
7	Enter an invalid password.	The password is entered and masked.
8	Enter a valid confirmation password.	The confirmation password is entered and masked.
9	Enter a valid phone number.	The phone number is entered.
10	Check the terms of service prompt.	The prompt is checked.
11	Click the submit button.	The form is submitted.
12	Assert the registration was unsuccessful and an appropriate alert message is displayed.	The registration failed and an alert message is displayed.
Test verdict	Failed	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed.
- The assertions are made using the *assert* command and its extensions.
- Command *execute script* is used to run a simple JS for a random string.
- The registration went through when it shouldn't have.

3.3.6 Test registration form with invalid confirmation password

Special Instructions

NONE

Test Case ID	TC_FUNCT_03_06	
Description	Test registration form with invalid confirmation password	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the "БИЛЕТИ" page.	The "БИЛЕТИ" page is loaded.
2	Verify the presence of registration button.	The registration button is present.
3	Click the registration button.	Open the registration page.
4	Fill the registration form.	The registration form has incorrect confirmation password.
5	Verify the presence of checkbox for confirmation the terms of use.	The checkbox is present.
6	Check the box.	
7	Verify there is a registration button.	The registration button is present.
8	Register a user	Click the register button.
9	Check if there is a text box with a warning.	The user is warned about the incorrect confirmation password.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the Registration button is present
- *assert check* is used to verify the checkbox is checked
- *store text* is used to get the content of the warning
- *execute script* is used to generate random names and emails
- *assert* is used to compare the one received and the expected one.

3.3.7 Test registration form with invalid phone

Special Instructions

NONE

Test Case ID	TC_FUNCT_03_07	
Description	Tests the registration of the tickets page with invalid phone	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the tickets page.	The tickets page is loaded.
2	Click the registration link.	Redirected to registration form.
3	Assert the registration form is displayed correctly.	The registration form is displayed correctly.
4	Enter a valid email.	The email is entered.
5	Enter a valid first name.	The first name is entered.
6	Enter a valid last name.	The last name is entered.
7	Enter a valid password.	The password is entered and masked.
8	Enter a valid confirmation password.	The confirmation password is entered and masked.
9	Enter an invalid phone number.	The phone number is entered.
10	Check the terms of service prompt.	The prompt is checked.
11	Click the submit button.	The form is submitted.
12	Assert the registration was unsuccessful and an appropriate alert message is displayed.	The registration failed and an alert message is displayed.
Test verdict	Passed	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed.
- The assertions are made using the *assert* command and its extensions.
- Command *execute script* is used to run a simple JS for a random string.

3.4 Traceability matrix

Test Case ID	Bug Description	Note
TC_FUNCT_03_05	The registration was successful with a password which doesn't follow the described rules.	The registration should fail with an appropriate message.

4. Test Tickets Login

4.1 Test Information

4.1.1 Test type

- ☒ Functional Test
☐ Performance Test

4.1.2 System Under Test

System name: BDZ Passenger Services

Version: 1.0

Short description of the system:

Online system which provides railway travel services like timetables, ticket reservation, live train tracing, news and information, etc.

4.1.3 Test Personnel

Name: Miroslav Dionisiev Date: 07-01-2022 Time/h: 0.5

Name: Pavel Sarlov Date: 07-01-2022 Time/h: 0.5

4.2 Test Summary

4.2.1 Results

Total number of test cases:	3
Total number of test cases passed:	3
Total number of test cases failed:	0
Total number of bugs found:	0

4.3 Test Cases

4.3.1 Test login form with valid data

Special Instructions

NONE

Test Case ID	TC_FUNCT_04_01	
Description	Test login form with valid data	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the "БИЛЕТИ" page.	The "БИЛЕТИ" page is loaded.
2	Verify the presence of login button.	The login button is present.
3	Click the login button.	Open the login page.
4	Fill the login form.	The login form is correctly filled.
5	Verify there is a login button.	The login button is present.
6	Click the login button	The "БИЛЕТИ" page is loaded
7	Click the name of the user.	Open the profile page.
8	Verify the user name.	The name on the page corresponds with the expected one.
Test verdict	Passed	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed.
- *assert element present* is used to assert whether the Login button is present
- *store text* is used to get the name of the logged user
- *assert* is used to compare the one received and the expected one.

4.3.2 Test login form with invalid email

Special Instructions

NONE

Test Case ID	TC_FUNCT_04_02	
Description	Tests the login of the tickets page with invalid email	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the tickets page.	The tickets page is loaded.
2	Click the login link.	Redirected to login form.
3	Assert the login form is displayed correctly.	The login form is displayed correctly.
4	Enter an invalid email.	The email is entered.
5	Enter a valid password.	The password is entered and masked.
6	Click the submit button.	The form is submitted.
7	Assert the login was unsuccessful and there is an appropriate message displayed.	The login failed and a message is displayed.
Test verdict	Passed	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed.
- The assertions are made using the *assert* command and its extensions.

4.3.3 Test login form with invalid password

Special Instructions

NONE

Test Case ID	TC_FUNCT_04_03	
Description	Test login form with valid data	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged out.	
Test Step ID	Test Step Description	Expected Result
1	Open the "БИЛЕТИ" page.	The "БИЛЕТИ" page is loaded.
2	Verify the presence of login button.	The login button is present.
3	Click the login button.	Open the login page.
4	Fill the login form.	The login form is filled with wrong password.
5	Verify there is a login button.	The login button is present.
6	Click the login button	A warning shows up
7	Check if there is a text box with a warning.	The user is warned about the incorrect username or password.
Test verdict	Passed	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed.
- *assert element present* is used to assert whether the Login button is present
- *store text* is used to get the warning text
- *assert* is used to compare the one received and the expected one.

4.4 Traceability matrix

Test Case ID	Bug Description	Note
--------------	-----------------	------

5. Test Tickets Reservation

5.1 Test Information

5.1.1 Test type

- ☒ Functional Test
☐ Performance Test

5.1.2 System Under Test

System name: BDZ Passenger Services

Version: 1.0

Short description of the system:

Online system which provides railway travel services like timetables, ticket reservation, live train tracing, news and information, etc.

5.1.3 Test Personnel

Name: Miroslav Dionisiev Date: 07-01-2022 Time/h: 0.5

Name: Pavel Sarlov Date: 07-01-2022 Time/h: 0.5

5.2 Test Summary

5.2.1 Results

Total number of test cases:	6
Total number of test cases passed:	6
Total number of test cases failed:	0
Total number of bugs found:	0

5.3 Test Cases

5.3.1 Test searching departure and arrival stations one-way ticket

Special Instructions

NONE

Test Case ID	TC_FUNCT_05_01	
Description	Tests searching one-way ticket of the tickets page	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the tickets page.	The tickets page is loaded.
2	Assert the search form is displayed correctly.	The search form is displayed correctly.
3	Enter a valid departure station.	The departure station is entered.
4	Choose the desired station from the dropdown list.	The station is selected.
5	Enter a valid arrival station.	The arrival station is entered.
6	Choose the desired station from the dropdown list.	The station is selected.
7	Enter a valid departure date.	The departure date is entered.
8	Assert there is no return date.	The return date is empty.
9	Click the submit button.	The search is submitted.
10	Assert the correct result is displayed.	The correct result is displayed.
Test verdict	Passed	

Comments:

- The assertions are made using the *assert* command and its extensions.
- The command *store* and its extensions are used to store helper variables.

5.3.2 Test searching departure and arrival stations two-way ticket

Special Instructions

NONE

Test Case ID	TC_FUNCT_05_02	
Description	Test searching departure and arrival stations two-way ticket	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the "БИЛЕТЫ" page.	The "БИЛЕТЫ" page is loaded.
2	The user logs in the system.	
3	Type the departure station name.	The station name is found dropdown list of stations.
4	Type the arrival station name	The station name is found dropdown list of stations.
5	A date of departure is chosen	
6	A date of return is chosen	
7	Verify presence of search button	The button is present
8	Click the search button	Loads two list for departure and return
9	Verify the names of the ticket list for both sections	The names are valid.
Test verdict	Passed	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed
- *assert element present* is used to assert button presence
- *store text* is used to get the names of the lists
- *execute script* is used to edit the stored text
- *assert* is used to compare the one received and the expected one.

5.3.3 Test searching without departure and arrival stations

Special Instructions

NONE

Test Case ID	TC_FUNCT_05_03	
Description	Tests searching with no stations entered.	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the tickets page.	The tickets page is loaded.
2	Assert the search form is displayed correctly.	The search form is displayed correctly.
3	Assert no departure station is selected.	The departure station field is empty.
4	Assert no arrival station is selected.	The arrival station field is empty.
9	Click the submit button.	The search is submitted.
10	Assert the search was unsuccessful and the fields are marked in red	The search failed and the fields are marked in red.
Test verdict	Passed	

Comments:

- The assertions are made using the *assert* command and its extensions.

5.3.4 Test ticket reservation for one person

Special Instructions

NONE

Test Case ID	TC_FUNCT_05_04	
Description	Test ticket reservation for one person	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the "БИЛЕТИ" page.	The "БИЛЕТИ" page is loaded.
2	The user logs in the system.	
3	Type the departure station name.	The station name is found dropdown list of stations.
4	Type the arrival station name	The station name is found dropdown list of stations.
5	A date of departure is chosen	
7	Verify presence of search button	The button is present
8	Click the search button	Loads two list for departure and return
9	Select a ticket option	
10	Verify presence of next step button	The button is present
11	Click the next step button	A personal data form loads
12	Fill the data	
13	Verify presence of next step button	The button is present
14	Click the next step button	A page to pick a place loads up
15	Verify presence of next step button	The button is present
16	Click the next step button	A payment page loads up
17	Verify the personal data for the ticket	The data is valid
Test verdict	Inconclusive	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed
- *assert element present* is used to assert button presence

- *store text* is used to get the names of the passenger and the traveling direction
- *execute script* is used to edit the stored text
- *assert* is used to compare the one received and the expected one.
- The test is inconclusive due to its incompleteness – the payment step was skipped.

5.3.5 Test ticket reservation for two people

Special Instructions

NONE

Test Case ID	TC_FUNCT_05_05	
Description	Tests the login page	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged in.	
Test Step ID	Test Step Description	Expected Result
1	Open the tickets page.	The tickets page is loaded.
2	Assert the search form is displayed correctly.	The search form is displayed correctly.
3	Enter a valid departure station.	The departure station is entered.
4	Choose the desired station from the dropdown list.	The station is selected.
5	Enter a valid arrival station.	The arrival station is entered.
6	Choose the desired station from the dropdown list.	The station is selected.
7	Enter a valid departure date.	The departure date is entered.
8	Assert there is no return date.	The return date is empty.
9	Click the submit button.	The search is submitted.
10	Wait for the list of results to be displayed.	A list of results is displayed.
11	Choose the first result of the displayed.	The first result is selected.
12	Click the next button.	Proceeds to the next form.
13	Wait for the passenger credentials form to be displayed.	The passenger credentials form is displayed.
14	Enter the necessary credentials for the first passenger.	The credentials are entered.
15	Click the button to add a second passenger.	A second passenger is added.
16	Wait for the passenger credentials form to be displayed.	The passenger credentials form is displayed.
17	Enter the necessary credentials for the second passenger.	The credentials are entered.
18	Click the next button.	Proceeds to the next form.
19	Wait for the seats form to be displayed.	The seats form is displayed.

20	Click the next button.	Proceeds to the next form.
21	Wait for the preview form to be displayed.	The preview form is displayed.
Test verdict	Inconclusive	

Comments:

- The assertions are made using the *assert* command and its extensions.
- The command *store* is used to store helper variables.
- The commands *do/repeat if* are used to loop and enter the credentials for the two passengers.
- The *wait* command and its extensions are used to give time to certain elements to load.
- The test is inconclusive due to its incompleteness – the payment step was skipped.

5.3.6 Test session time expiration

Special Instructions

NONE

Test Case ID	TC_FUNCT_05_06	
Description	Test session time expiration	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	The user should be logged in.	
Test Step ID	Test Step Description	Expected Result
1	Open the "БИЛЕТЫ" page.	The "БИЛЕТЫ" page is loaded.
2	The user logs in the system.	
3	Type the departure station name.	The station name is found dropdown list of stations.
4	Type the arrival station name	The station name is found dropdown list of stations.
5	A date of departure is chosen	
7	Verify presence of search button	The button is present
8	Click the search button	Loads two list for departure and return
9	Wait for the session to expire	
10	Verify the expiration	The is a message for the expiration
Test verdict	Passed	

Comments:

- Commands *store xpath count* and *if* are used to check if there is a login link. If there isn't, the user is logged in and will be logged out so that the test can proceed
- *assert element present* is used to assert button presence
- *store text* is used to get the warning for session expiration
- *assert* is used to compare the one received and the expected one.

5.4 Traceability matrix

Test Case ID	Bug Description	Note
--------------	-----------------	------

6. Test Live

6.1 Test Information

6.1.1 Test type

- ☒ Functional Test
☐ Performance Test

6.1.2 System Under Test

System name: BDZ Passenger Services

Version: 1.0

Short description of the system:

Online system which provides railway travel services like timetables, ticket reservation, live train tracing, news and information, etc.

6.1.3 Test Personnel

Name: Miroslav Dionisiev Date: 07-01-2022 Time/h: 0.5

Name: Pavel Sarlov Date: 07-01-2022 Time/h: 0.5

6.2 Test Summary

6.2.1 Results

Total number of test cases:	3
Total number of test cases passed:	3
Total number of test cases failed:	0
Total number of bugs found:	0

6.3 Test Cases

6.3.1 Test live tracer with valid departure station

Special Instructions

NONE

Test Case ID	TC_FUNCT_06_01	
Description	Test live tracer with valid departure station	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the "Live" page.	The "Live" page is loaded.
2	Type the name of the station.	The station name is found dropdown list of stations.
3	Verify the presence of "Заминавщи" button.	The button is present.
4	Click the "Заминавщи" button.	A page with list of trains loads.
5	Verify that the list is with departing trains	The content of the list meets the criteria.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the "Заминавщи" button is present
- *store text* is used to get the title of the list with trains
- *assert* is used to compare the one received and the expected one.

6.3.2 Test live tracer with valid arrival station

Special Instructions

NONE

Test Case ID	TC_FUNCT_06_02	
Description	Tests the live tracer page with valid arrival station	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	None	
Test Step ID	Test Step Description	Expected Result
1	Open the live tracer page.	The live tracer page is loaded.
2	Assert there is a search box for stations.	The search box is present.
3	Enter a valid arrival station.	The arrival station is entered.
4	Wait for the dropdown menu to be displayed.	The dropdown is displayed.
5	Click the desired station option.	The station is selected.
6	Assert the correct station results are displayed	The correct results are displayed.
Test verdict	Passed	

Comments:

- The assertions are made using the *assert* command and its extensions.
- The command *store* is used to store helper variables.
- The *wait* command and its extensions are used to give time to certain elements to load.

6.3.3 Test live tracer with invalid station

Special Instructions

NONE

Test Case ID	TC_FUNCT_06_03	
Description	Tests the live tracer page with invalid station	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions		
Test Step ID	Test Step Description	Expected Result
1	Open the live tracer page.	The live tracer page is loaded.
2	Assert there is a search box for stations.	The search box is present.
3	Enter an invalid arrival station.	The arrival station is entered.
4	Wait for the dropdown menu to be displayed.	The dropdown is not displayed.
5	Click Enter on the keyboard.	A return action is sent to the form.
6	Assert there have been no changes.	No changes occurred.
Test verdict	Passed	

Comments:

- The assertions are made using the *assert* command and its extensions.
- The command *store* is used to store helper variables.
- The *wait* command and its extensions are used to give time to certain elements to load.

6.4 Traceability matrix

Test Case ID	Bug Description	Note
--------------	-----------------	------

7. Test Radar

7.1 Test Information

7.1.1 Test type

- ☒ Functional Test
☐ Performance Test

7.1.2 System Under Test

System name: BDZ Passenger Services

Version: 1.0

Short description of the system:

Online system which provides railway travel services like timetables, ticket reservation, live train tracing, news and information, etc.

7.1.3 Test Personnel

Name: Miroslav Dionisiev Date: 07-01-2022 Time/h: 0.5

Name: Pavel Sarlov Date: 07-01-2022 Time/h: 0.5

7.2 Test Summary

7.2.1 Results

Total number of test cases:	2
Total number of test cases passed:	2
Total number of test cases failed:	0
Total number of bugs found:	0

7.3 Test Cases

7.3.1 Test radar with active train and its route

Special Instructions

NONE

Test Case ID	TC_FUNCT_07_01	
Description	Test radar with active train and its route	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	There should be at least one active train.	
Test Step ID	Test Step Description	Expected Result
1	Open the “Радар” page.	The “Радар” page is loaded.
2	Verify the presence of a drop down menu button.	The button is present.
3	Click over a train from the menu	The route of the train loads
4	Verify the number of the train.	The train number is valid.
Test verdict	Passed	

Comments:

- *assert element present* is used to assert whether the drop down menu button is present
- *store text* is used to get the numbers of the clicked train and the one of the load route
- *assert* is used to compare the one received and the expected one.

7.3.2 Test radar with train number

Special Instructions

NONE

Test Case ID	TC_FUNCT_07_02	
Description	Tests the radar page with train number	
Applicable for	Firefox, Edge, Chrome	
Initial Conditions	There should be at least one active train.	
Test Step ID	Test Step Description	Expected Result
1	Open the radar page.	The radar page is loaded.
2	Assert the interactable map is displayed correctly.	The map is displayed correctly.
3	Click on the arrow button and wait for the side panel to be displayed.	The side panel is displayed.
4	Enter a valid train number in the filter box.	The train number is entered and the list gets filtered.
5	Assert the train with the entered number exists.	The train with the given number exists.
6	Click the train with the entered number.	A pop-up with the train schedule info is displayed.
Test verdict	Passed	

Comments:

- The assertions are made using the *assert* command and its extensions.
- The *wait* command and its extensions are used to give time to certain elements to load.

7.4 Traceability matrix

Test Case ID	Bug Description	Note
--------------	-----------------	------