

Testlink - opis realizacji

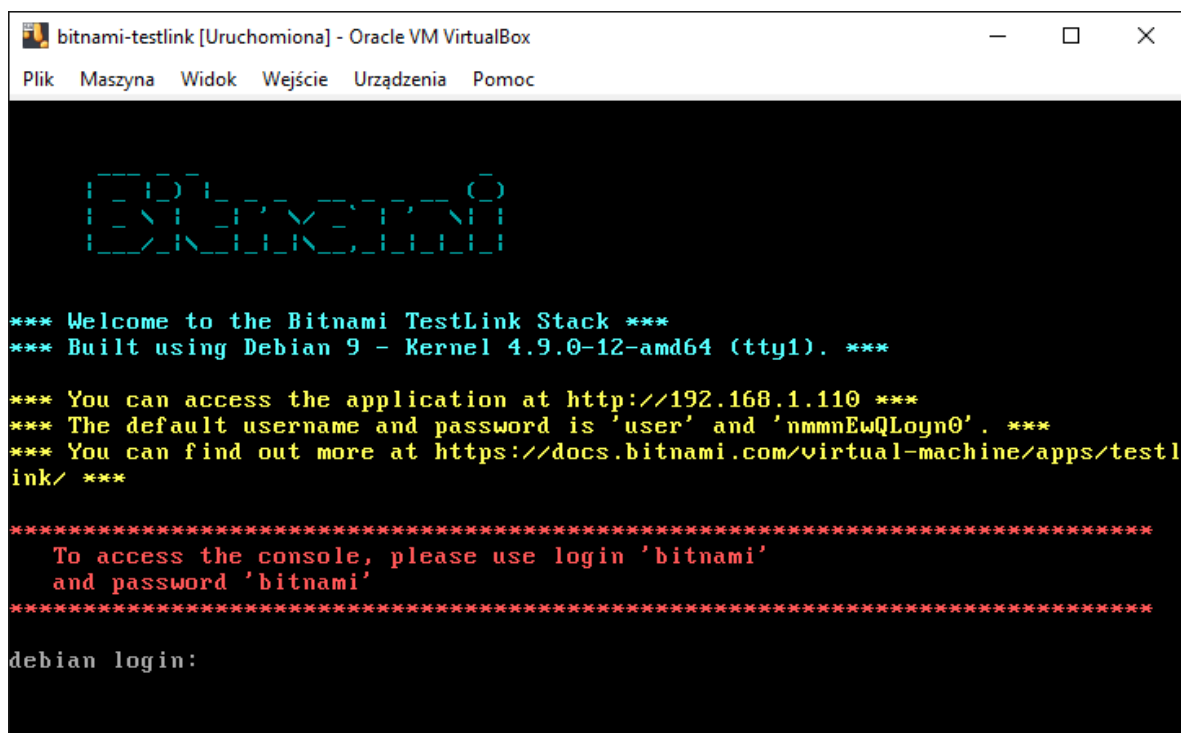
27 maja 2020

Spis treści

1	Uruchomienie obrazu Bitnami	2
2	Konfiguracja projektu	4
3	Testy manualne	5
3.1	Testy ewaluacji ruchów	5
3.1.1	Test Case 1: 2-literowy ruch	6
3.1.2	Test Case 2: 5-literowy ruch	6
3.2	Testy dodawania punktów użytkownika	7
3.2.1	Test Case 1: Po dwóch ruchach	7
3.2.2	Test Case 2: Po czterech ruchach	8
3.3	Testy dodawania sztucznej inteligencji do rozgrywki	8
3.3.1	Test Case 1: Na początku	8
3.4	Testy dodawania liter na planszę	9
3.4.1	Test Case 1: 2-literowy ruch	9
3.5	Specyfikacja testów	9
3.6	Wykonanie testów	10
4	Testy automatyczne	11
5	Testy Selenium	13

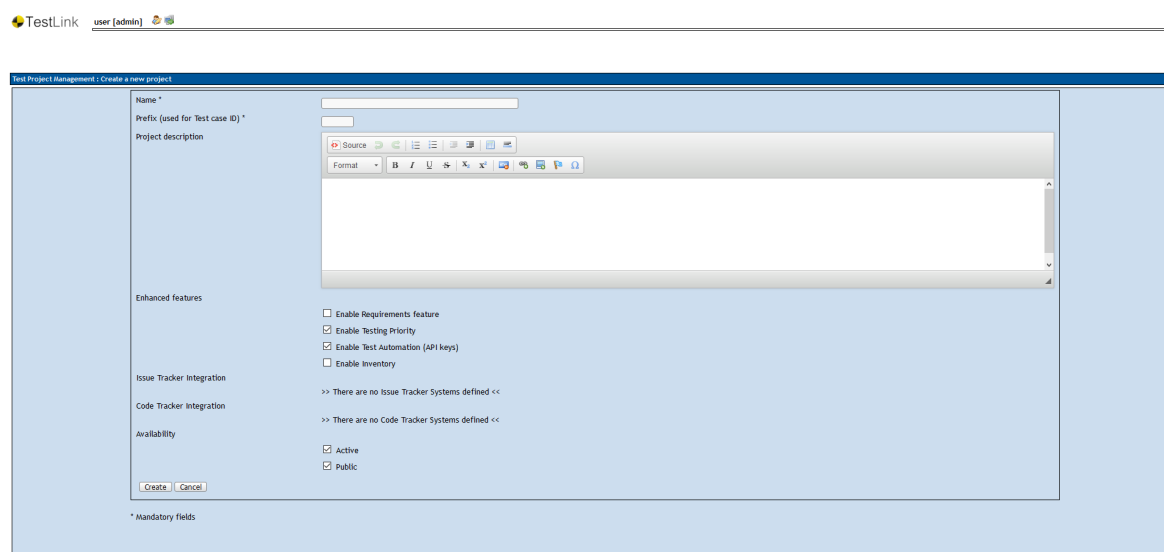
1 Uruchomienie obrazu Bitnami

Obraz z rozszerzeniem .ova został pobrany [stąd](#) i zaimportowany do programu *Virtual Box*. Po dokonanej konfiguracji sieciowej i uruchomieniu - ponownym, bo pierwsze nie przyniosło pożądanych efektów - pokazał się taki ekran:



Rysunek 1: Ekran startowy Bitnami

Po przejściu do aplikacji webowej na hoście zalogowano się na konto z uprawnieniami administratora:



Rysunek 2: Panel administratora

```
C:\WINDOWS\system32\cmd.exe
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Połączenie lokalne* 2:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . : fe80::fd63:72fe:aede:50e5%13
IPv4 Address. . . . . : 192.168.1.108
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1

Ethernet adapter Połączenie sieciowe Bluetooth:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Ethernet adapter vEthernet (Default Switch):

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . : fe80::6594:5c35:31c3:e981%26
IPv4 Address. . . . . : 192.168.255.177
Subnet Mask . . . . . : 255.255.255.240
Default Gateway . . . . . :

C:\Users\Radek>
```

Rysunek 3: Konfiguracja sieciowa na *host*

```
radek@radek-VirtualBox: ~
radek@radek-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.109 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::d917:98df:bc0f:e779 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:fa:e3:41 txqueuelen 1000 (Ethernet)
    RX packets 37452 bytes 48047445 (48.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 7705 bytes 1624110 (1.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 773 bytes 85496 (85.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 773 bytes 85496 (85.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

radek@radek-VirtualBox:~$ g
```

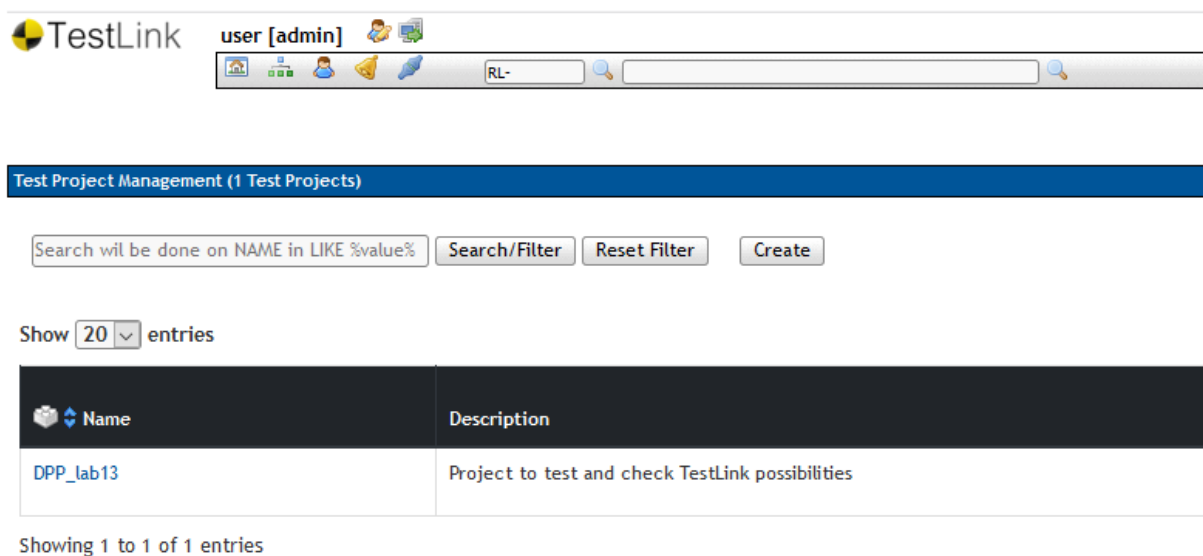
Rysunek 4: Konfiguracja sieciowa na *guest2*

Czyli ostatecznie osiągnięto taką konfigurację:

- host (Windows 10) - 192.168.1.108,
- guest1 (Bitnami-Testlink) - 192.168.1.110,
- guest2 (Ubuntu 20.04) - 192.168.1.109

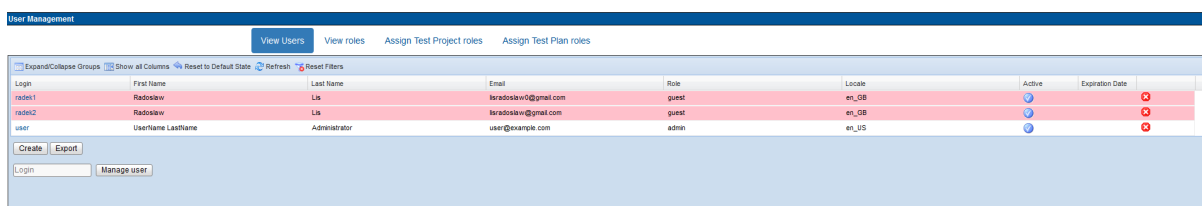
Do Testlinka można było zalogować się zarówno na *host* jak i na *guest2*.

2 Konfiguracja projektu



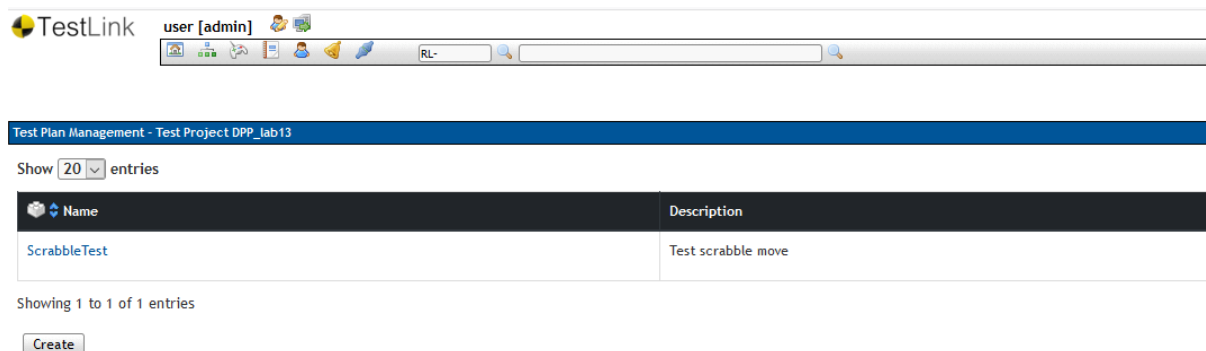
Rysunek 5: Utworzenie przykładowego projektu w panelu administratora

Poprzez gościa na Ubuntu zostało dodanych 2 przykładowych użytkowników:



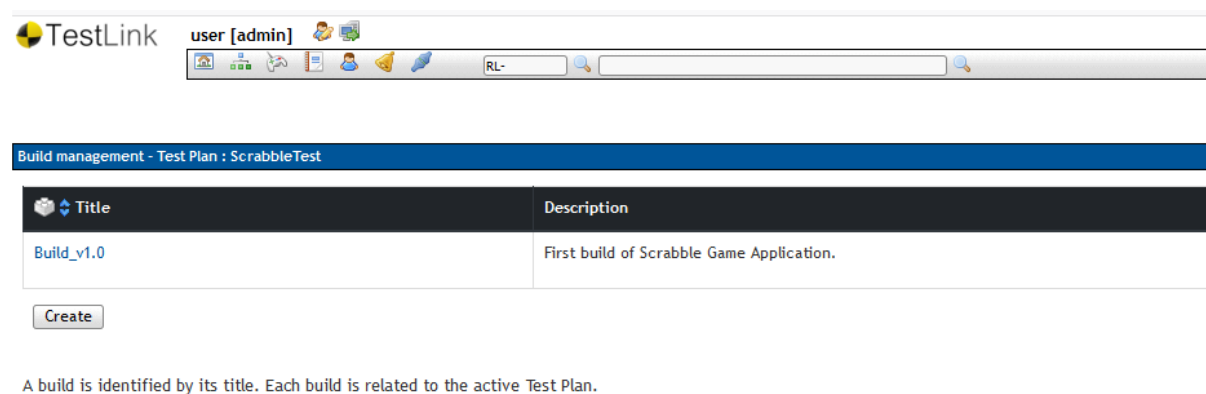
Rysunek 6: Lista użytkowników

W panelu *Test Plan Management* został utworzony przykładowy plan testów:



Rysunek 7: Utworzony plan testów

Utworzony został również pierwszy *build/release* przypisany do utworzonego wcześniej planu (wersja v1.0):



Rysunek 8: Pierwszy build

3 Testy manualne

Aplikacja wykorzystana do testów to Flask’owa aplikacja z lab7 umożliwiająca grę w Scrabble kilku osobom (oraz możliwość dołączenia do gry sztucznej inteligencji).

3.1 Testy ewaluacji ruchów

Zostały zdefiniowane 2 przypadki manualnych testów w celu sprawdzenia poprawności funkcji liczenia wykonanego przez użytkownika ruchu.

3.1.1 Test Case 1: 2-literowy ruch

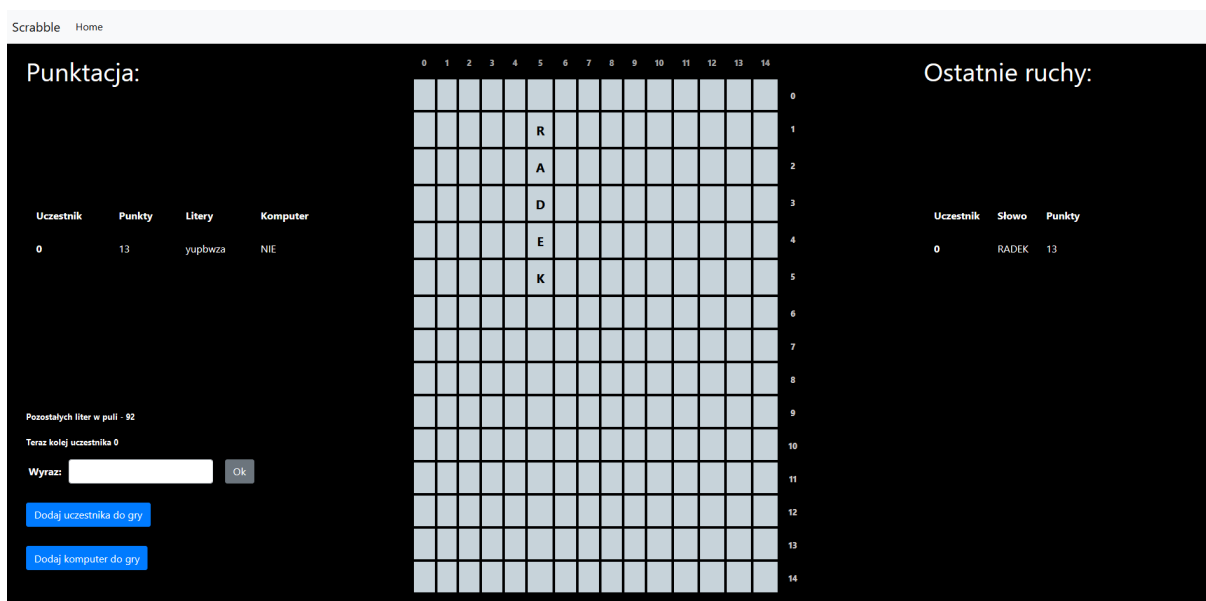
Test Case			
RL-1: Check simple two-letter move - Version1			
Created on 05/27/2020 12:15:53 by user Last modified on 05/27/2020 12:38:40 by user			
Summary The test case for check correctness of evaluating simple two-letter move			
Preconditions 1. The user should be connected to the internet. 2. The user should have cloned application from https://git.e-science.pl/ris241385_dpp/ris_flaskapi 3. The user should locally run application by typing "flask run" in command line			
#	Step actions	Expected Results	Execution
1	Open http://localhost:5000/game/view website	The website should be opened and the board should be visible	Manual ✖ ✔
2	Add user to the game ("Dodaj uczestnika do gry" button)	User should be added and info about him should be visible in the "Punkacja" section	Manual ✖ ✔
3	Make the move by typing in "Wyraz" text-field move "4:5:1;4:6:S" and click Ok	Move should be added to game log	Manual ✖ ✔
4	Check value of the move made in game log section	Value of the move made should be 2 (- 1, S - 1)	Manual ✖ ✔
Create step Resequenece Steps			
Status : Draft Importance : Medium Execution type : Manual <input type="checkbox"/> Apply To All Steps			
Estimated exec. (min) : <input type="text"/> Save			
Keywords : None			
Platforms : None			

Rysunek 9: Zdefiniowane poszczególne kroki Test Case 1

3.1.2 Test Case 2: 5-literowy ruch

Test Case			
RL-2: Check five-letter move - Version1			
Created on 05/27/2020 12:41:23 by user Last modified on 05/27/2020 12:44:45 by user			
Summary			
Preconditions 1. The user should be connected to the internet. 2. The user should have cloned application from https://git.e-science.pl/ris241385_dpp/ris_flaskapi 3. The user should locally run application by typing "flask run" in command line			
#	Step actions	Expected Results	Execution
1	Open http://localhost:5000/game/view website	The website should be opened and the board should be visible	Manual ✖ ✔
2	Add user to the game ("Dodaj uczestnika do gry" button)	User should be added and info about him should be visible in the "Punkacja" section	Manual ✖ ✔
3	Make the move by typing in "Wyraz" text-field move "1:5:R;2:5:A;3:5:D;4:5:E;5:5:K" and click Ok	Move should be added to game log	Manual ✖ ✔
4	Move should be added to game log	Value of the move made should be 13	Manual ✖ ✔
Create step Resequenece Steps			
Status : Draft Importance : Medium Execution type : Manual <input type="checkbox"/> Apply To All Steps			
Estimated exec. (min) : <input type="text"/> Save			
Keywords : None			
Platforms : None			

Rysunek 10: Zdefiniowane poszczególne kroki Test Case 2



Rysunek 11: Widok, który powinien ujrzeć tester po wykonaniu Test Case 2

3.2 Testy dodawania punktów użytkownika

Zostały zdefiniowane 2 przypadki manualnych testów w celu sprawdzenia poprawności podliczania punktów uczestnika w rozgrywce.

3.2.1 Test Case 1: Po dwóch ruchach

Test Case				
RL-3 : Check user points after 2 moves - Version1				
Summary				
Testing correctness of adding user points in 2 moves.				
Preconditions				
1. The user should be connected to the internet.				
2. The user should have cloned application from https://git.e-science.pl/ris/241385_dpp/ris_flaskapi				
3. The user should locally run application by typing "flask run" in command line				
Step actions	Expected Results	Execution		
1 Open http://localhost:5000/game/view website	The website should be opened and the board should be visible	Manual	✗	+
2 Add user to the game ("Dodaj uczestnika do gry" button)	User should be added and info about him should be visible in the "Punkcja" section	Manual	✗	+
3 Make the move by typing in "Wyraz" text-field move "4:5:1:4:6:S" and click Ok	Move should be added to game log	Manual	✗	+
4 Make another move by typing in "Wyraz" text-field move "6:6:0:5:6:R" and click Ok	Move should be added to game log	Manual	✗	+
5 Check user points in "Punkcja" section in the column "Punkty"	Value in column "Punkty" should be 5	Manual	✗	+
<input type="button" value="Create step"/> <input type="button" value="Resequenece Steps"/>				
Status : <input type="text" value="Draft"/> Importance : <input type="text" value="Medium"/> Execution type : <input type="text" value="Manual"/> <input type="checkbox"/> Apply To All Steps Estimated exec. (min) : <input type="text"/> <input type="button" value="Save"/> Keywords : <input type="text" value="None"/> Platforms : <input type="text" value="None"/>				

Rysunek 12: Zdefiniowane poszczególne kroki Test Case 1

3.2.2 Test Case 2: Po czterech ruchach

Test Case				
RL-4 : Check user points after 4 moves - Version1				
Summary				
Testing correctness of adding user points in 4 moves.				
Preconditions				
1. The user should be connected to the internet. 2. The user should have cloned application from https://git.e-science.pl/ris241385_dpp/ris_flaskapi 3. The user should locally run application by typing "flask run" in command line				
#	Step actions	Expected Results	Execution	
1	Open http://localhost:5000/game/view website	The website should be opened and the board should be visible	Manual	✖ ✔
2	Add user to the game ("Dodaj uczestnika do gry" button)	User should be added and info about him should be visible in the "Punktacja" section	Manual	✖ ✔
3	Make the move by typing in "Wyraz" text-field move "4:5:I,4:6:S" and click Ok	Move should be added to game log	Manual	✖ ✔
4	Make another move by typing in "Wyraz" text-field move "6:6:O;5:6:R" and click Ok	Move should be added to game log	Manual	✖ ✔
5	Make another move by typing in "Wyraz" text-field move "1:5:R;2:5:A;3:5:D;4:5:E;5:5:K" and click Ok	Move should be added to game log	Manual	✖ ✔
6	Make another move by typing in "Wyraz" text-field move "1:1:R;1:2:O" and click Ok	Move should be added to game log	Manual	✖ ✔
7	Check user points in "Punktacja" section in the column "Punkty"	Value in column "Punkty" should be 21	Manual	✖ ✔
<button>Create step</button> <button>Resequene Steps</button>				

Rysunek 13: Zdefiniowane poszczególne kroki Test Case 2

3.3 Testy dodawania sztucznej inteligencji do rozgrywki

Został zdefiniowany jeden manualny test w celu sprawdzenia poprawności dodawania sztucznej inteligencji do gry.

3.3.1 Test Case 1: Na początku

Test Case				
RL-5 : Add at the begin - Version1				
Summary				
Testing correctness of adding artificial intelligence at the beggining of the game.				
Preconditions				
1. The user should be connected to the internet. 2. The user should have cloned application from https://git.e-science.pl/ris241385_dpp/ris_flaskapi 3. The user should locally run application by typing "flask run" in command line				
#	Step actions	Expected Results	Execution	
1	Open http://localhost:5000/game/view website	The website should be opened and the board should be visible	Manual	✖ ✔
2	Add AI to the game ("Dodaj komputer do gry" button)	AI should be added and info about it should be visible in the "Punktacja" section	Manual	✖ ✔
<button>Create step</button> <button>Resequene Steps</button>				
Status : Draft Importance : Medium Execution type : Manual <input type="checkbox"/> Apply To All Steps				
Estimated exec. (min) : <input type="text"/> <button>Save</button>				
Keywords : None				
Platforms : None				

Rysunek 14: Zdefiniowane poszczególne kroki Test Case 1

3.4 Testy dodawania liter na planszę

3.4.1 Test Case 1: 2-literowy ruch

Test Case			
RL-6: Check visibility after first move - Version1			
Summary			
Testing correctness of adding letters to the board after first move.			
Preconditions			
1. The user should be connected to the internet. 2. The user should have cloned application from https://git.e-science.pl/r1s241385_dpp/r1s_flaskapi 3. The user should locally run application by typing "flask run" in command line			
Step actions	Expected Results	Execution	
1. Open http://localhost:5000/game/view website	The website should be opened and the board should be visible	Manual	✗
2. Add user to the game ("Dodaj uczestnika do gry" button)	User should be added and info about him should be visible in the "Punktacja" section	Manual	✗
3. Make the move by typing in "Wyraz" text-field move "4:5;4:6:S" and click Ok	Letters should be putted on the board this way: 4 row, 5 column - I 4 row, 6 column - S	Manual	✗

Status: Importance: Execution type: ☐ Apply To All Steps

Estimated exec. (min):

Keywords:

Platforms:

Rysunek 15: Zdefiniowane poszczególne kroki Test Case 1

3.5 Specyfikacja testów

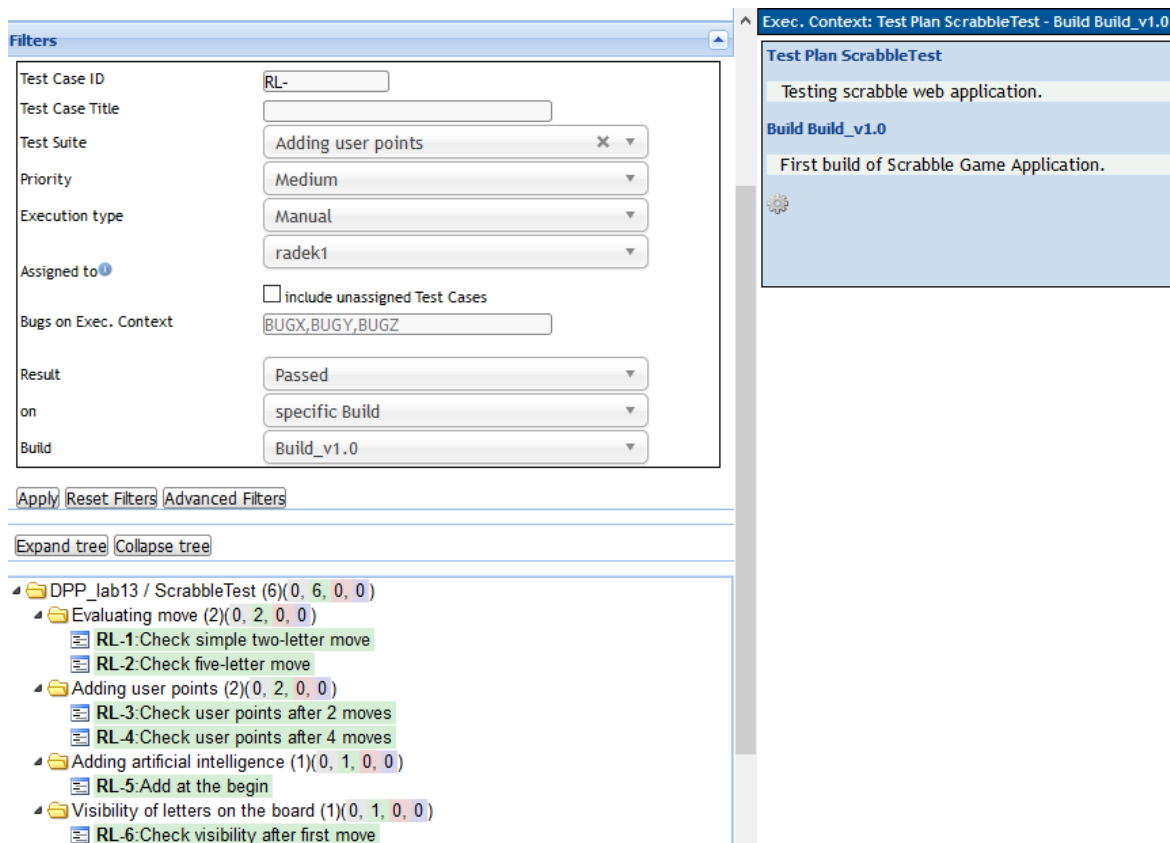
Po zdefiniowaniu powyższych testów złożyły się one w następującą specyfikację:

- 📁 DPP_lab13 (6)
 - 📁 Evaluating move (2)
 - 📁 Adding user points (2)
 - RL-3:Check user points after 2 moves
 - RL-4:Check user points after 4 moves
 - 📁 Adding artificial intelligence (1)
 - RL-5:Add at the begin
 - 📁 Visibility of letters on the board (1)
 - RL-6:Check visibility after first move

Rysunek 16: Zdefiniowane poszczególne kroki Test Case 1

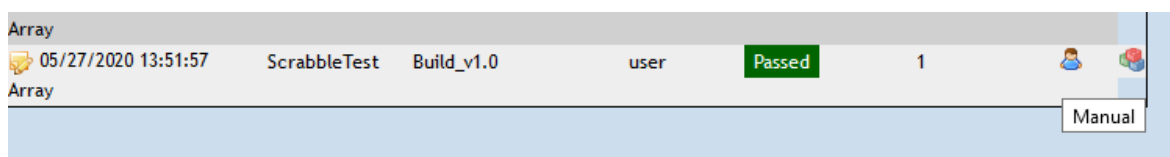
3.6 Wykonanie testów

Testy zostały przypisane do utworzonego planu i do konkretnego użytkownika na *guest2* i wykonane w sekcji *Test execution* na Ubuntu. Postępując zgodnie ze zdefiniowanymi krokami w każdym teście wszystkie przeszły i zostały oznaczone jako *passed*:



Rysunek 17: Status testów po wykonaniu

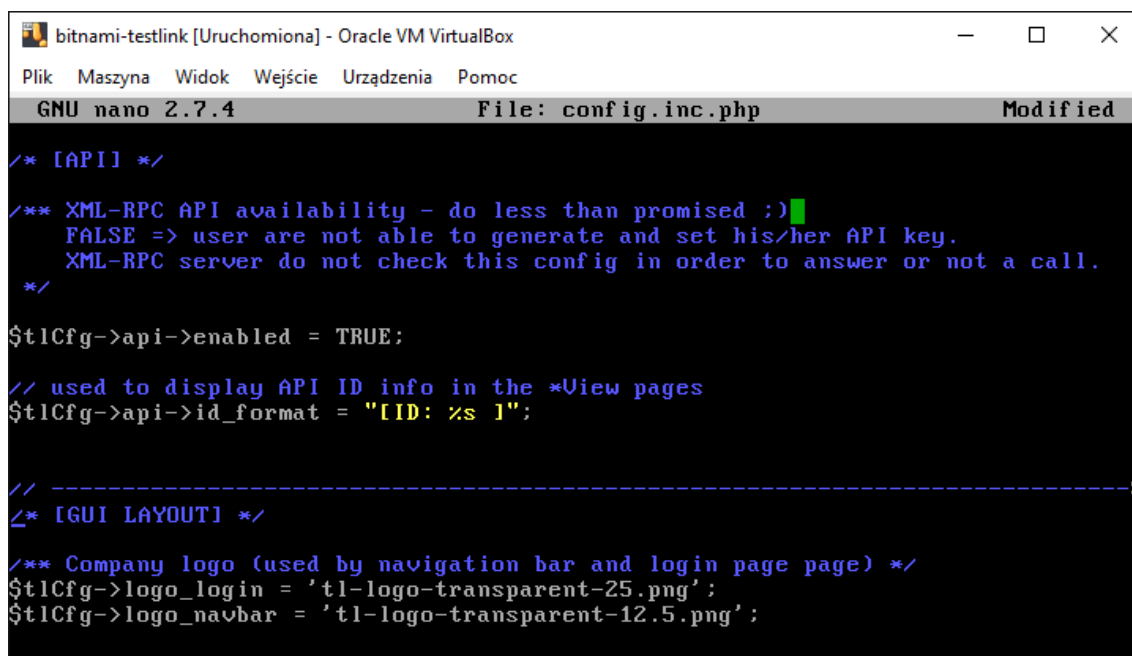
Plany testów są do zobaczenia w pliku *ManualTestsPlan.xml*, a rezultat ich wykonania w *ExecutedManualTests.xml*. Ponadto wszystkie testy są wyeksportowane do pliku *DPP_lab13.testproject - deep.xml*.



Rysunek 18: Pozytywny status jednego z testów

4 Testy automatyczne

Testy automatyczne zostały przeprowadzone na aplikacji wyszukującej optymalny ruch w grze w Scrabble z lab7. Odblokowany został interfejs API:



```
bitnami-testlink [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
GNU nano 2.7.4 File: config.inc.php Modified

/* [API] */

/** XML-RPC API availability - do less than promised :)
  FALSE => user are not able to generate and set his/her API key.
  XML-RPC server do not check this config in order to answer or not a call.
 */

$t1Cfg->api->enabled = TRUE;

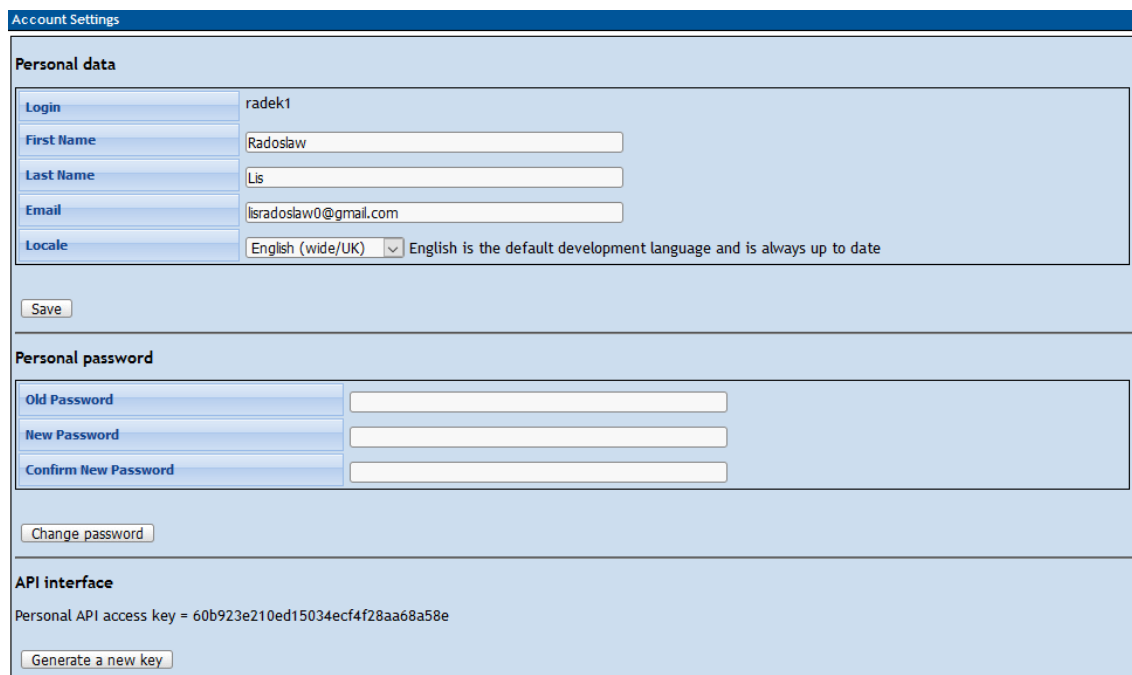
// used to display API ID info in the *View pages
$t1Cfg->api->id_format = "[ID: %s ]";

// -----
/* [GUI LAYOUT] */

/** Company logo (used by navigation bar and login page page) */
$t1Cfg->logo_login = 'tl-logo-transparent-25.png';
$t1Cfg->logo_navbar = 'tl-logo-transparent-12.5.png';
```

Rysunek 19: Zawartość pliku *config.inc.php*

Został też wygenerowany klucz API dla użytkownika *radek1* na *guest2*:



Account Settings

Personal data

Login	radek1
First Name	Radoslaw
Last Name	Lis
Email	lsradoslaw0@gmail.com
Locale	English (wide/UK) English is the default development language and is always up to date

Save

Personal password

Old Password	
New Password	
Confirm New Password	

Change password

API interface

Personal API access key = 60b923e210ed15034ecf4f28aa68a58e

Generate a new key

Rysunek 20: Wygenerowany klucz API

Został zdefiniowany jeden test automatyczny:

Test Case

Warning! This Test Case version has been executed.

RL-10 : Check whether length of anagrams is correct - Version1

Summary

Testing correctness of length of list of anagrams as a result from find_anagrams functionality with given letters.

Preconditions

1. The user should have Python 3.7 installed

#	Step actions	Expected Results	Execution
1	Open wordsENG.txt file	File opened	Manual
2	Make trie from words	Trie made and stored in cache	Manual
3	Invoke find_anagrams method with "abc" letters	Get list of anagrams	Manual
4	Check length of given list	Length of the list should be 11	Manual

Create step

Resequence Steps

Status : Draft Importance : Medium Execution type : Manual ☐ Apply To All Steps

Estimated exec. (min) :

Save

Keywords : None

Platforms : None

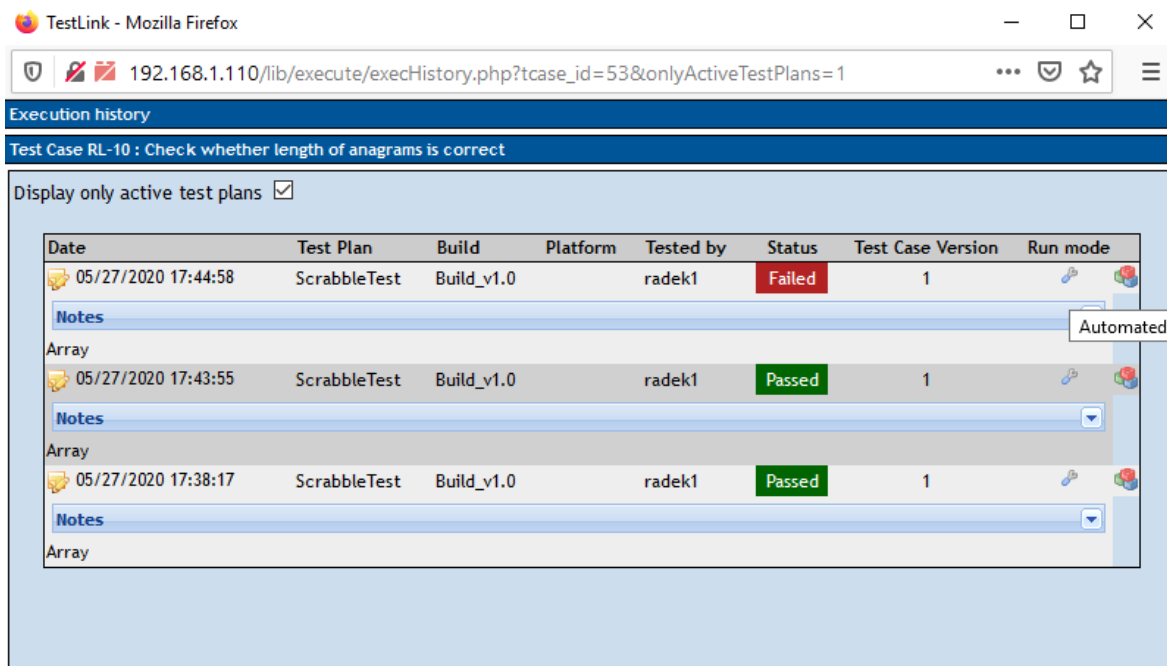
Rysunek 21: Test poprawności generowanych anagramów

W Pythonie został napisany kod (wykorzystano kod z instrukcji), który miał przeprowadzić test, połączyć się z Te-stlinkiem i wysłać raport. Kod znajduje się w pliku `automatic_test.py`, a plik `.oxps` to `ExecutedAutomaticTest.oxps`.

Test Suite : Anagrams correctness

Test Case RL-10: Check whether length of anagrams is correct [Version : 1]			
Summary:			
Testing correctness of length of list of anagrams as a result from find_anagrams functionality with given letters.			
#	Step actions	Expected Results	Execution Status
1	Open wordsENG.txt file	File opened	Passed
2	Make trie from words	Trie made and stored in cache	Passed
3	Invoke find_anagrams method with "abc" letters	Get list of anagrams	Passed
4	Check length of given list	Length of the list should be 11	Failed
Execution type:			
Manual			
Estimated exec. duration (min):			
Priority:			
Medium			
Execution Details			
Build			
Build_v1.0			
Tester			
radek1			
Execution Result:			
Failed			
Execution Mode:			
Automated			
Execution duration (min):			
Execution notes			
Failed			

Rysunek 22: Rezultat (negatywny) egzekucji testu



Rysunek 23: Historia egzekucji testu

5 Testy Selenium

Został również przeprowadzony jeden automatyczny test za pomocą Selenium na aplikacji webowej, na której były przeprowadzane testy manualne. Kod znajduje się w pliku *selenium_test.py*, a plik *.oxps* to *ExecutedSelenium-Test.oxps*.



Rysunek 24: Poszczególne kroki testu

Execution history							
Test Case RL-11 : Selenium test							
Display only active test plans <input checked="" type="checkbox"/>							
Date	Test Plan	Build	Platform	Tested by	Status	Test Case Version	Run mode
05/27/2020 19:13:19	ScrabbleTest	Build_v1.0		radek1	Passed	1	Automated
Notes							
Array							

Rysunek 25: Pozytywny przebieg testu

Expand tree	Collapse tree
<ul style="list-style-type: none"> DPP_lab13 (8) <ul style="list-style-type: none"> Evaluating move (2) <ul style="list-style-type: none"> RL-1:Check simple two-letter move RL-2:Check five-letter move Adding user points (2) <ul style="list-style-type: none"> RL-3:Check user points after 2 moves RL-4:Check user points after 4 moves Adding artificial intelligence (1) <ul style="list-style-type: none"> RL-5:Add at the begin Visibility of letters on the board (1) <ul style="list-style-type: none"> RL-6:Check visibility after first move Anagrams correctness (1) <ul style="list-style-type: none"> RL-10:Check whether length of anagrams is correct Executing move (1) <ul style="list-style-type: none"> RL-11:Selenium test 	

Rysunek 26: Ostateczna konfiguracja testów