Movierama Test Plan

Author: Grigoris Kavadias
(gregory.kavadias@gmail.com)

Table of Contents

1. Introduction

In the current document, the project plan [R2] for the *MovieRama* web application (http://workable-movierama.herokuapp.com/) testing work to be done is defined, including the test cases covering the application functionality, in accordance with [R1]. Movierama provides information on movies currently played in theaters and also incorporates a search mechanism, as well as pagination functionality. The structuring of the test cases is perceived within a unified scenario, that covers the given acceptance criteria. Additional negative scenarios were provided when necessary.

The current document is expected to guide reader's thinking in order to cover the full functionality specified in [R1] and also serves as a basis for building the automated scripts.

It is noted that no special attention was paid to the underlying Movierama architecture and its information sources.

For authenticity purposes, the current document was created from scratch based on an empty template.

2. Test Scenario

In this scenario, all Test Cases of the assignment are included. In addition, it is noted that Public user is the common actor and there are no special requirements necessary for accessing the web application, other than Internet connectivity.

2.1 TC01:View and navigation among movies list

Short Description:Public user views and navigates through a list of movies played the current week in the theaters

	Step Description	**Expected Result**
1	When accessing *MovieRama* web site for the first time, verify the following: ● The default number of items / page is 5 ● At the top of the page, there exists an input field for searching the website ● Just below the search box, there exists a number displaying the total number of movies ● There exist links to all *MovieRama* pages in the following form: “1 2 3 … Last page” ● For each item (movie) the following are available: ○ Title (in bold) ○ Year of production (below the title) ○ Movie Description ○ Starring actors ○ Number of reviews	The layout, information and its structure is as expected Note: keep a note of the “movies in theaters this week” in order to refer to it later on
2	Make sure that none of the displayed items has more than 5 actors	There’s no item with more than 5 actos
3	Move down to the bottom of the page and confirm that since you’re on the first page there exists “next” link to navigate you to the next page but there	There exists “next” but no “previous” link

	exist no “previous” link	
4	Modify the number of items to be displayed per page by selecting the value “10” from the “Page size” drop down box	The number of items per page is now 10
5	Verify that at the top of the page, the number of “movies in theaters this week” is the same as that in step #1	The number of “movies in theaters this week” is the same as that in step #1
6	Repeat steps 3-4 for the remaining options in the drop down list, defining the items to be displayed per page (20,30,50)	Layout is OK for the rest of the options and also the number of “movies in theaters this week” is the same as that in step #1
7	Navigate from the first page towards the last page by using the “1 2 3 … Last page”	● The last page is reached without a problem ● The number displaying the total number of movies is the same as that on step #1 on every page visited
8	Being on the last page, navigate towards the first page by using the “1 2 3 … Last page”	● The last page is reached without a problem ● The number displaying the total number of movies is the same as that on step #1
9	Repeat steps 7-8, this time by using the “next”, “previous” buttons	FIrst and last pages are accessed without a problem
10	Since the URL follows the pattern: http://workable-movierama.herokuapp.com/page/<Page Number> try accessing the page (malformed URL, big inexistent page number is inserted) :http://workable-movierama.herokuapp.com/page/999999	A message should inform the user by displaying a message such as: “undefined movies in theaters this week”
11	Repeat step #10 this time by malforming the URL by adding latin characters http://workable-movierama.herokuapp.com/page/xxxxxx	A message should inform the user by displaying a message such as: “undefined movies in theaters this week” or an HTTP 404 related error message
12	Repeat step #10 this time by malforming the URL adding special characters:	A message should inform the user by displaying a message such as: “undefined movies in theaters this week” or an HTTP

	http://workable-movierama.herokuapp.com/page/%%%5E&##	404 related error message By no means, should the portal freeze, runtime exception be thrown etc

2.3 TC02: MovieRama search

Short Description:Public user searches for movies titles on *MovieRama* web application. In addition, correct pagination of the search results, navigation among them and variations of the number of the returned items per page are verified. Finally the “AND” operator is examined for searching more than one words in the movie title

	Step Description	Expected Result
1	Set page size to 50 for better visibility of the search results	Page size is set to 50
2	Fill in “Bomb” in the input field for searching the website and click “Enter’	Search results of movies including “Bomb” in the title are retrieved
3	Check that the “# results found” below the input field for searching pertains to the number of search results	Total search results math the “# results found”
4	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
5	Fill in “Tom” in the input field for searching the website and click “Enter” Note: In this case the results are much more than in the case of “Bomb”	Search results of movies including “Tom” in the title are retrieved
6	Check that the “# results found” below the input field for searching pertains to the number of search results	Search results math the “# results found”
7	Set page size of the search results to 5	Page size is set to 5
8	Confirm that the “# results found” is the same as that found at step #5 and has not erroneously been modified due to the page size	“# results found” is the same as that found at step #5
9	Navigate among search results from the first page towards the last page by using the “1 2 3 … Last page” and	● The last page is reached without a problem ● “# results found” is the same as that

	check that the titles contain the word: “tom”	found at step #5 ● “tom” is included in the results
10	Being on the last page, navigate towards the first page by using the “1 2 3 … Last page”	● The last page is reached without a problem ● “# results found” is the same as that found at step #5
11	Repeat steps 7-8, this time by using the “next”, “previous” buttons	● FIrst and last pages are accessed without a problem ● “# results found” is the same as that found at step #5
12	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
13	Search for movies including the word: “jijikos” in their title	“A message 0 results found” is displayed to the end user
14	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
15	Search for movies including the word: “Γρηγόρης” in their title (ensure that search works for Unicode)	“A message 0 results found” is displayed to the end user
16	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
17	Search for movies including the word: “ingl**é**s” in their title (ensure that search works for Unicode)	Movies matching the Spanish word ingl**é**s in their titles are retrieved
18	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
19	Search for the following special characters: “#&^”	Make sure that the following message is displayed to the end user: “A message 0 results found” is displayed to the end user and **the application does not malfunction**
20	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved

21	Fill in “Tom*” in the input field for searching the website and click “Enter’	Taking into account that <u>no “*” wildcard</u> was defined in [R1] Make sure that the application does not malfunction and should return no rows if there are no matches
22	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
23	Fill in “*[1]Tom” in the input field for searching the website and click “Enter’	Taking into account that <u>no “*” wildcard</u> was defined in [R1] Make sure that the application does not malfunction and should return no rows if there are no matches
24	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
25	Fill in any of the following words: “**?**Tom”, “**$**Tom”, “Tom**?**”, “Tom**$**”, in the input field for searching the website and click “Enter’	Make sure that no malfunction occurs and the search mechanism works as expected, i.e. as if these special characters were normal characters
26	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
27	Fill in “Tom Jerry” in the input field for searching the website and click “Enter’	Search results of movies including Tom AND Jerry in the title are retrieved
28	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
29	Fill in “Jerry Tom” in the input field for searching the website and click “Enter’	The same search results found in step #27 are retrieved
30	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
31	Repeat the steps above for the keywords: “lie”, “good”	Results of the “good lie”, “lie good” are identical
32	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved

[1] In the specifications, no reference to wildcard characters *, ?

32	Click on ⊗ icon to clean up the previous search	Search field is cleaned up and the *MovieRama* home page is retrieved
38	Search for “Tom Jerry Movie”	The movie: “Tom and Jerry - The Movie” is retrieved Note: It is assumed that the current movie is in theaters the current week. Otherwise, the message “0 results found” should be displayed

3. Non Functional testing

In this chapter, various considerations are taken into account with respect to non functional testing. More precisely, since the application is available on the web, it is accessible by various devices and also, there exists a performance requirement according to which the system should respond within 3 seconds.

3.1 Performance requirements

The system should respond within 3 seconds. In order to assure that the system responses will have to be within the timeframe of 3 seconds.

3.1 Access the application via various devices and web browsers[2]

The various functional test cases of section: Test Scenario, should be followed on various devices and web browsers, to assure that the layout, pagination and functionality work as expected in various environments.

More precisely, the following basic configurations should be used:

- Desktop machine
  - Mozilla Firefox
  - IE 8,9,10
  - Google Chrome
  - Opera
- Tablets
  - ipad (iOS 8.x)
  - Android tablet (4.3 or later)
- Mobile Phones
  - iPhone (iOS 8.x)
  - Android mobile phones

[2] The exact devices set, as well as browsers per device should be further specified

4. Automated Scripts

Junit4 and Selenium Webdriver API[3] have been used with Java 7. The main reason was that the same scripts can be easily executed in more than one browsers (e.g. Internet Explorer, Google Chrome etc).

The scripts should be executed as Junit Tests from within Eclipse and more precisely the scripts are the following;

- *MovieRamaJUnitTestSuite*: The TestSuite including the following Test Cases:
  - *TC01*: Unit Test aiming to verify that there's no problem with accessing the MovieRama website
  - *TC02*: Unit Test dealing with 1 negative and 1 positive search test. More precisely in the context of the negative test, a search for the word "jijikos" is performed whereas in the positive test, the word "Tom' is searched
- *MovieRamaUtilities*: java class including constants and methods common in both test cases

Installation Instructions

- Setup ECLIPSE ide with Java 7.
- Download locally the “**MovieRama.zip**” file accessible on: “https://drive.google.com/?authuser=0#folders/0B3JHMQ6nml5AZzEtLVlhT1RKemc” google drive folder.
- In order to run the scripts as Unit Tests, in ECLIPSE do the following: to import the “MovieRama” project: “File > Import > General > Existing Projects into workspace” and select the “**MovieRama.zip**” file.
- After successfully importing the project, run the *MovieRamaJUnitTestSuite.*

[3] http://docs.seleniumhq.org/docs/03_webdriver.jsp

References

1. Specifications of the current assignment (AssignmentfortheQAAutomationEngineer.pdf) [R1]
2. ISTQB EXAM CERTIFICATION [R2] (http://istqbexamcertification.com/what-is-the-purpose-and-importance-of-test-plans/)