**TEST Design**

**Table of Contents**

[Introduction: 3](#_Toc5495069)

[1.1 Objectives 3](#_Toc5495070)

[Overview: 3](#_Toc5495071)

[Prerequisites: 3](#_Toc5495072)

[Page Object Model: 3](#_Toc5495073)

[Design Plan: 3](#_Toc5495074)

[4.1 Phase 1: 3](#_Toc5495075)

[4.2 Changes in Phase 1: 4](#_Toc5495076)

[4.2 Phase 2: 4](#_Toc5495077)

[4.3 Final Implementation: 5](#_Toc5495078)

[Notes: 5](#_Toc5495079)

[Acknowledgement: 5](#_Toc5495080)

# **Introduction:**

## 1.1 Objectives

This document will cover the details of what functions are present and how each function will work and how the application will be tested.

# **Overview:**

* The script will be written in Java using Selenium 3.3.1, with an object-oriented approach
* Project will have two packages
* First package – namely upGradAssignment \_Imdb\_FetchSort - will have source code files.
* File Names: “FetchMovieInformation.java”,”CreateExcel.java” and “SortingMetrics.java”
* Second package – namely upGradAssignment\_Imdb\_CommonLocators - will have Common Element Locators for the “Landing page and Top 250 Movies” page
* File Names: “IMDb\_CommonLocators.java” and “SortByMetrics.java”
* “FetchMovieInformation” will act as a base file for “SortByMetrics” and the latter will use the functions from the former to verify sorting metrics

# **Prerequisites:**

For Phase 1 and 2:

* Selenium (3.3.1) and latest chromedriver needs to be downloaded in the system
* JXL and Apache POI needs to be downloaded
* Project needs to be built by adding all the above dependencies and libraries manually

# **Page Object Model:**

* Page Object Model will be implemented
* “upGradAssignment\_Imdb\_CommonLocators” package will have the common locators for two pages, Landing page and the Information page
* “Imdb\_LandingPage\_CommonLocators” will have the objects for identifying the common locators on landing page
* “Imdb\_SortingPage\_CommonLocators” will have the objects for identifying the common locators on page to test sorting metrics
* All the source code methods will point to these pages wherever necessary

# **Design Plan:**

## 4.1 Phase 1:

**FetchMovieInformation.java**

Functions:

1. launchChrome()
   1. Will launch the chrome browser
   2. Has return type “WebDriver”
   3. Set Property needs to be modified to user’s chromedriver path
2. launchIMDBTop250Page()
   1. Will navigate to landing page and then navigate to the IMDBTop250 page
   2. On this page, the no of items in the Movie Table will be retrieved
3. storeMovieTitles()
   1. Movie titles will be fetched
   2. Will write the same to excel
4. storeMovieRelease()
   1. Will fetch the movie release date information
   2. Will write the same to excel
5. storeMovieRatings()
   1. Will fetch IMDb movie ratings
   2. Will write the same to excel
6. main()
   1. Will initialize the WebDriver variable “driver” with launchChrome() function
   2. Will call all the above functions to be executed

## 4.2 Changes in Phase 1:

1. CreateExcel.java has been added
   1. This has the necessary Apache POI libraries necessary to write in excel
2. setBorder()
   1. This sets the border for the heading in excel
3. setColor()
   1. Sets the color for the heading cells in excel

## 4.2 Phase 2:

**SortingMetrics.java**

Functions:

1. launchChrome()
   1. Will launch the chrome browser
   2. Has return type “WebDriver”
   3. Set Property needs to be modified to user’s chromedriver path
2. dropdown()
   1. Will click the dropdown
   2. Will fetch all the necessary sorting metrics
3. sortOrderButton()
   1. Will click the ascending/descending order button
4. sortBase()
   1. Will have the base logic to verify the sorting metrics.
   2. Will verify both ascending and descend order
   3. Will verify sorting in case there is a clash in the sorting items
   4. Parameters, as in the factors on whose basis the sorting needs to be carried out will be passed to this base file
5. sortByRanking()
   1. Will have the parameters needed to be passed to the base file
   2. Will refer to “FetchMovieInformation” for necessary stored information
   3. Will have the additional logic needed to help the base file
6. sortByImdbRating()
   1. Will have the parameters needed to be passed to the base file
   2. Will refer to “FetchMovieInformation” for necessary stored information
   3. Will have the additional logic needed to help the base file
7. sortByReleaseDate()
   1. Will have the parameters needed to be passed to the base file
   2. Will refer to “FetchMovieInformation” for necessary stored information
   3. Will have the additional logic needed to help the base file
8. sortByNoOfRatings()
   1. Will have the parameters needed to be passed to the base file
   2. Will refer to “FetchMovieInformation” for necessary stored information
   3. Will have the additional logic needed to help the base file
9. main()
   1. Will initialize the WebDriver variable “driver” with launchChrome() function
   2. Will call all the above functions to be executed

## 4.3 Final Implementation:

With the help of build tool, the “Prerequisites” for phase 1 and phase 2 will be skipped, and minimal setup will be required.

Prerequisite: The client machine will need maven setup

Integration with CI and Build tool will be carried out. Need reference work for implementing design and development for the same, as discussed per the test plan.

# **Notes:**

Please note that this is a rough design, and it might change in case there is additional technical requirement or some logical changes need to be done which might be identified during the implementation phase. Any changes will be informed at the time of further deliveries

# **Acknowledgement:**

The assignment will be carried out based on this test design. Development will start, assuming that the design plan works for upGrad recruitment team.

|  |  |
| --- | --- |
| Name: | Siddhant Thakur |
| Title: | Sr. Quality Engineer |
| Date: | 5th April 2019 |