**Application The Movie DB**

Test Plan

**1. Introduction**

This Test Plan document is intended to document the information needed to plan and control the validation tests of Application The Movie DB - V.01.00. The document describes the general test plan for a basic search for information about a movie. The Test Cases, where the general plan is described, will be executed to validate the application.

**1.1 Scope**

This document describes the Test Plan, to be used by Application The Movie DB - V.01.00 project, to evaluate the functional quality, reliability and performance. The following tests will be performed: black box, exploratory, usability and automation of tests.

**2. Planning for the tests**

**2.1** Approach to testing:

|  |  |
| --- | --- |
| **Functional Requirement** | **Non-Functional Requirement** |
| **RF01 -** The app should allow the search for a registered movie | **RNF01 -** The app must be developed in the Java language |
| **RF02 -** The app should display the listing of all registered movies, by category and in alphabetical order, making it easy to find the movie for the user | **RNF02 -** The database used should be SQLite |
| **RF03 -** The app should display all the information about the registered movies | **RNF03 -** The app must be compatible with Android 6.0 (Marshmallow) onwards. |
| **RF04 -** The app should display the trailer for all registered movies | **RNF04 -** The app must have a Responsive Design. |

**2.2** Detail of the test approach:

|  |  |
| --- | --- |
| **Type of Test:** | Functional |
| **Test Subtype:** | Requirements |
| **Purpose of Test:** | Test the search functionality of a movie entered in the database |
| **Testing Requirements:** | RF01 -The app should allow the search for a registered movie |

|  |  |
| --- | --- |
| **Type of Test:** | Functional |
| **Test Subtype:** | Requirements |
| **Purpose of Test:** | Test the functionality of the movie list, it should be displayed by category and in alphabetical order |
| **Testing Requirements:** | RF02 -The app should display the listing of all registered movies, by category and in alphabetical order, making it easy to find the movie for the user. |

|  |  |
| --- | --- |
| **Type of Test:** | Functional |
| **Test Subtype:** | Requirements |
| **Purpose of Test:** | Test the features of viewing the information for each movie, when selected. |
| **Testing Requirements:** | RF03 -The app should display all the information about the registered movies |

|  |  |
| --- | --- |
| **Type of Test:** | Functional |
| **Test Subtype:** | Requirements |
| **Purpose of Test:** | Test video functionality when selecting a movie |
| **Testing Requirements:** | RF04 -The app should display the trailer for all registered movies |

|  |  |
| --- | --- |
| **Type of Test:** | Non Functional |
| **Test Subtype:** | Implementation |
| **Purpose of Test:** | Test whether the system is encoded in the Java Language |
| **Testing Requirements:** | RNF01 -The app must be developed in the Java language |

|  |  |
| --- | --- |
| **Type of Test:** | Non Functional |
| **Test Subtype:** | Implementation |
| **Purpose of Test:** | Test and verify that the system is making use of the requested database |
| **Testing Requirements:** | RNF02 -The database used should be SQLite |

|  |  |
| --- | --- |
| **Type of Test:** | Non Functional |
| **Test Subtype:** | Usability |
| **Purpose of Test:** | Test if app is compatible with Android 6.0 and new models |
| **Testing Requirements:** | RNF03 -The app must be compatible with Android 6.0 (Marshmallow) onwards. |

|  |  |
| --- | --- |
| **Type of Test:** | Non Functional |
| **Test Subtype:** | Usability |
| **Purpose of Test:** | Test whether the app is misaligned when accessed by a device or tablet with the highest resolution screen. |
| **Testing Requirements:** | RNF04 -The app must have a Responsive Design. |

**3.0 Tests Cases**

|  |  |
| --- | --- |
| **UC001: Search movies** | |
| **Steps:** | **Expected result:** |
| 1. Access the "Application The Movie DB" 2. Type the name of a movie in the search field 3. Click the "Magnifying glass" icon | 1. The home screen should be displayed 2. The field should receive the scanned data 3. The application should display the movie, with its respective information |

|  |  |
| --- | --- |
| **FE01: Movie not found** | |
| **Steps:** | **Expected result:** |
| 1. Access the "Application The Movie DB" 2. Type the name of a movie in the search field 3. Click the "Magnifying glass" icon | 1. The home screen should be displayed 2. The field should receive the scanned data 3. The application should display the message "Movie not found" |

|  |  |
| --- | --- |
| **FE02: Invalid field** | |
| **Steps:** | **Expected result:** |
| 1. Access the "Application The Movie DB" 2. Type the name of a movie in the search field 3. Click the "Magnifying glass" icon without typing data | 1. The home screen should be displayed 2. The field should receive the scanned data 3. The application should display the message "Invalid field" |

|  |  |
| --- | --- |
| **FE03: Movie unavailable** | |
| **Steps:** | **Expected result:** |
| 1. Access the "Application The Movie DB" 2. Type the name of a movie in the search field 3. Click the "Magnifying glass" icon | 1. The home screen should be displayed 2. The field should receive the scanned data 3. The application should display the message "Movie unavailable" |

|  |  |
| --- | --- |
| **UC002: Movie list** | |
| **Steps:** | **Expected result:** |
| 1. Access Acsses the "Application The Movie DB" 2. Scroll the movie list | 1. The home screen should be displayed 2. The app should display the movies listed by category and in alphabetical order |

|  |  |
| --- | --- |
| **UC003: Movie info** | |
| **Steps:** | **Expected result:** |
| 1. Access the "Application The Movie DB" 2. Select a movie from the list | 1. The home screen should be displayed 2. The app should display all information about the movie |

|  |  |
| --- | --- |
| **FE01: Movie without information** | |
| **Steps:** | **Expected result:** |
| 1. Access the "Application The Movie DB" 2. Select a movie from the list | 1. The home screen should be displayed 2. The application should display the message " Information not found " |

|  |  |
| --- | --- |
| **UC004: Movie trailer** | |
| **Steps:** | **Expected result:** |
| 1. Access the "Application The Movie DB" 2. Select a movie from the list 3. Select the "Play" button | 1. The home screen should be displayed 2. The application should display all information about the movie along with the trailer 3. The app should display the movie trailer |

|  |  |
| --- | --- |
| **FE01: Pause in the trailer** | |
| **Steps:** | **Expected result:** |
| 1. Access the "Application The Movie DB" 2. Select a movie from the list 3. Select the "Play" button 4. Select the "Pause" button | 1. The home screen should be displayed 2. The application should display all information about the movie along with the trailer 3. The app should display the movie trailer 4. The application should stop the video whenever the button is selected |

|  |  |
| --- | --- |
| **FE02: Reboot the trailer** | |
| **Steps:** | **Expected result:** |
| 1. Access the "Application The Movie DB" 2. Select a movie from the list 3. Select the "Play" button 4. Select the "Pause" button 5. Select the "Play" button again | 1. The home screen should be displayed 2. The application should display all information about the movie along with the trailer 3. The app should display the movie trailer 4. The application should stop the video whenever the button is selected 5. The app should continue to play the video, from where it was paused |

**3. Bug Reports**

During the validation of the app, I found some bugs, which I will report below:

* Model: G570M / Version: Android 7.0
* Model: A500M / Version: Android 6.0.1

**FE01: Movie not found**

When performing search for a non-registered movie, the app does not display the message "Movie not found", displaying a black screen and then returns to the home screen.

Test carried out on 2 more samples of the same models mentioned above, with the same error persisting in both.

**UC002: Movie list**

Application does not display movie categories in alphabetical order, making it difficult to find movies for the user.

Test carried out on 2 more samples of the same models mentioned above, with the same error persisting in both.

**UC003: Movie info**

The app displays special characters in the text. Ex: All words that have an accent or ç, are replaced by an interrogation (?), leaving the spelling of the text incorrect and making it difficult for the user to read.

Test carried out on 2 more samples of the same models mentioned above, with the same error persisting in both.

**FE01: Pause in the trailer**

The application only pauses the video after 3 seconds the button has been selected, not allowing the user to pause the video at the moment he really wants.

Test carried out on 2 more samples of the same models mentioned above, with the same error persisting in both.

**FE02: Reboot the trailer**

Application gave force close to restart a video making it impossible for the user to continue watching the movie trailer.

Test carried out on 2 more samples of the same models mentioned above, with the same error persisting

**4. Test Reports**

4.1 Tests done

* Black box test:

Ensure the quality of the app, by checking the output of the data using inputs of various types.

* Exploratory test

Ensure the quality of the app, thus avoiding unpleasant surprises in relation to the final result of the application.

* Usability test

Ensure the quality of the app, verifying the ease of the app, being properly understood and manipulated by the user.

* Reliability Testing

The inputs, outputs and operations of the requirements defined for the application are validated, ensuring that the application receives the data correctly, performing the processing and the result correctly

* Performance testing

Analyze the execution flow, response time and operational limits.

**Tools used:**

* TestLink: Run the test cases
* Jira: Open and close issues

**In the 2nd sprint, two more features should be developed, which are:**

* Movie Sharing: via WhatsApp and via E-Mail
* List of interest: User you can create a list of movies of your interest, to facilitate your search for movies within the app

**Conclusion**

After testing and all issues fixed, Sprint can be delivered in 2 weeks, thereby ensuring the quality of the app and delivering a simple and user-friendly app for the user.