SE 3XA3: Test Plan
Shuffle

Team 16, ASK Studios
Aidan Schonewille, schonea
Sullivan Stobo, stobos
Karlo Delalic, delalik

December 6, 2017

Contents

Table 1: **Revision History**

Date	**Version**	**Notes**
10/27/2017	1.0	Filled in all sections
December 5	1.1	Finalized

List of Tables

1 General Information

1.1 Purpose

The purpose of the project Shuffle is provide an easy way to shuffle play music videos based on a specific keyword from Youtube.

1.2 Scope

This project uses JavaScript utilizing Reactjs and various YouTube APIs. The scope of our testing will cover mainly the correctness of our usage of the various YouTube APIs, any Javascript algorithms that define functionality and the UI made using React. Reliability and robustness will be paid special attention to. We will be doing both black-box and white-box testing.

1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations**

Abbreviation	Definition
JS	JavaScript
API	Application Programming Interface
UI	User Interface
React	React.js

Table 3: **Table of Definitions**

Term	Definition
React.js	A component-based JavaScript library for creating user interfaces.
YouTube	A web service for uploading, sharing and playing video.
Shuffle	The title of the original project and also the title of our project.

1.4 Overview of Document

This document will outline the testing process to be used to verify the web application 'Shuffle.' The test will be categorized into two categories; System Tests, which will test the requirements of the system and Proof of concept tests, which will test the functionality of the system. Every test will contain a description of how it will be carried out and by whom.

2 Plan

2.1 Software Description

Shuffle is a web app that generates personalized playlists based on the users input and preferences. It acts as a alternative to the major music streaming services.

2.2 Test Team

The following project members are responisble for all procedures of the validation process including writing and executing tests:

- Aidan Schonewille
- Sullivan Stobo
- Karlo Delalic

2.3 Automated Testing Approach

Shuffle will be tested using both automated and human testing. Program functions and the UI will be tested using an automated testing tool. The majority of the testing will be done using automated testing tools with an exception of testing the correctness of the video. The Test Team will compile a list of search terms, and it will check if the video is a good match (e.g. if it is a music video and not any other type such as an interview).

2.4 Testing Tools

Jest will be used to test the various React components and vanilla JS for robustness and correctness to avoid any potential bugs or errors that we may have missed. Manual testing will be used for the UI as it is impossible to automatically conform that every UI element is in the correct location. Manual testing will also be used for visual confirmationis required for correctness.

2.5 Testing Schedule

View our Gantt chart here.

3 System Test Description

3.1 Tests for Functional Requirements

3.1.1 Testing for correctness

The product shall allows music videos to be streamed from YouTube based on a search query and when one video ends another video relevant to the query shall begin playing.

Basic functionality test

1. test-id1

   Type: Functional

   Initial State: The Webpage has just loaded.

   Input: A search query

   Output: A music video shall begin playing

   How test will be performed: A member of the test team shall observe whether or not a music video plays.

2. test-id2

   Type: Functional

   Initial State: A video has just ended

Input: The search query previously used to find the last video

Output: Another relevant video

How test will be performed: A member of the test team shall observe whether or not a new relevant video plays.

3.1.2 Testing video control

The product shall allow the user to pause the current video, go the next video or to back to the previous video

Next, previous and play tests.

1. test-id1

   Type: Functional

   Initial State: The Webpage has just loaded and a search query has been provided. A video is playing.

   Input: Clicking the pause button

   Output: The current video stops playing

   How test will be performed: A member of the test team shall observe whether or not the video stops playing.

2. test-id2

   Type: Functional

   Initial State: The Webpage has just loaded and a search query has been provided. A video is playing.

   Input: The next button is pressed and then the previous button is pressed

   Output: A new video should begin to play and then the video the was previously playing should begin to play

   How test will be performed: A member of the test team shall observe whether or not when the next button is pressed a new video plays and when the previous button is pressed the previous video plays.

3.2 Tests for Nonfunctional Requirements

3.2.1 Testing for usability

A minimal interface should be used as not to obstruct the video being played or distract the user. The product shall fit on a single page and be intuitive to use.

Look, feel and usability testing

1. test-id1

   Type: Functional

   Initial State: The webpage has just loaded.

   Input/Condition: None

   Output/Result: The webpage shall be completely viewable with scrolling up or down

   How test will be performed: A member of the test team shall confirm that the webpage can be entirely view without scrolling.

2. test-id2

   Type: Functional

   Initial State: A video is playing on the webpage

   Input: None

   Output: No text should be interfering or overlapping with the video

   How test will be performed: A member of the test team shall observe the webpage and for any overlapping text.

3.2.2 Testing for Performance

The product shall preform similarly to the previous 'Shuffle' project in term of response time. The project shall run in most HTML 5 environments.

Performance and Operational tests

1. test-id1

   Type: Manual

   Initial State: The webpage has not loaded

   Input/Condition: A test team member will attempt to open the project on Chrome, Firefox and Edge

   Output/Result: The project should successfully open on all three browser in a similar amount of time.

   How test will be performed: A member of the test team shall observe the time it takes the webpage to load and compare each result to each other.

2. test-id2

   Type: Manual

   Initial State: The webpage has not loaded

   Input: A test team member shall open the original 'Shuffle' project concurrently with ours.

   Output: Our project should open in a similar amount of time or faster.

   How test will be performed: A member of the test team shall observe the time it takes both webpages to open.

4 Tests for Proof of Concept

4.1 Testing for Visual Correctness

Basic Functionality Test

1. test-id1

   Type: Functional

   Initial State: Site has not been loaded.

Input: Opening site.

Output: Video player renders.

How test will be performed: Site will be opened to check if the player initializes.

2. test-id2

   Type: Functional

   Initial State: Site has not been loaded.

   Input: Opening site and specifying a search term "Radiohead".

   Output: Player loads and plays a random Radiohead music video.

   How test will be performed: A search term will be added manually in the code and the site will be loaded.

3. test-id3

   Type: Functional

   Initial State: Site is playing a Radiohead music video.

   Input: Opening site and specifying a search term "Radiohead".

   Output: Player loads and plays a random Radiohead music video.

   How test will be performed: A search term will be added manually in the code and the site will be loaded.

4.2 Testing for Internal Functions

Basic Functionality Test

1. test-id1
   Type: Manual

   Initial State: Site is offline.

   Input: Starting site with search term "Radiohead".

   Output: Player loads and plays a random Radiohead music video.

   How test will be performed: A search term will be added manually in the code and the list of videos to play will be outputted to the console. This will be run multiple times to check for randomness.

2. test-id2
   Type: Manual

   Initial State: Site is offline.

   Input: Starting site with search term "".

   Output: Player loads and creates a playlist of random music videos.

   How test will be performed: A blank search term will be added manually in the code and the list of videos to play will be outputted to the console. This will be run multiple times to check for randomness.

5 Comparison to Existing Implementation

Three tests will be made to compare Shuffle to the existing implementation:

- Test **3.2.1, id2** will be used to test the usability and responsiveness of Shuffle and the existing implementation
- Test **3.2.2, id1** will be used to test the performance of Shuffle and the existing implementation
- Test **3.2.1, id1** will be used to test the overall satisfaction of Shuffle and the existing implementation

6 Unit Testing Plan

Jest will be used for React component and vanilla JS testing, and manual testing will be used for UI testing.

6.1 Unit testing of internal functions

In order create unit tests for the internal functions of the program, certain methods that return values can be tested. This will involve taking the methods and giving them input values. Given what they are supposed to output and that they actually output a series of unit tests can be created. Unit tests will include unit tests that contain proper inputs and inputs that generate exceptions. Jest will be used for this purpose. Test cases will be created for each function and Jest will take care of the testing.

6.2 Unit testing of output files

In order to create test cases for the UI, the non-functional requirements will need to be taken into account. Examples of such are: the video to search term match, type of video returned, functionality of buttons, search bars, and other UI updating components. Manual testing will be used to test the UI as human verification of the video is required as not enough information is included in the video to distinguish from an incorrect match.