Test Plan

Aman Khoja

David Mathew

Minah Popal

Rohan Kozhikunnathu Samuel

Justin Yim

Group 12

ABSTRACT:

The consumption of media via video streaming services is higher than ever. With video streaming platforms such as Netflix, Twitch, and YouTube growing continuously, consumers of these platforms have an endless amount of content to watch. Oftentimes, consumers who may watch multiple video media or platforms at once, have to change between multiple screens or browser windows. Proposed is a multimedia platform which will allow consumers to watch multiple video streaming services at once, on a single screen, to alleviate the need to switch between screens or windows.

Table Of Contents

LIST OF TABLES:

INTRODUCTION:

This Test Plan document shall provide an in-depth description of all immediately foreseeable test cases. Each system level test case will be detailed with a description, preconditions, assumptions, the testing steps, and the expected results. The traceability of each test case back to its appropriate use case shall be apparent and clear. Furthermore, this document shall explore different techniques for generating each test case and shall provide the criteria on which to judge the quality of each test case.

REQUIREMENTS / SPECIFICATIONS-BASED SYSTEM LEVEL TEST CASES:

Test Case 1 (Table 1)

Title:	Login Page
Description:	User shall be able to input their username and password to enter into the website
Precondition:	The user must already have a account with our service though the registration page
Assumption:	User already has a account set up with the service
Test Steps:	1. Lands on the homepage of the website 2. Click on “Login” button 3. Enter username 4. Enter password 5. Click enter
Expected Results:	Upon filling out there username and password the customer shall be able to enter into the site and access the services of the site

Test Case 2 (Table 2)

Title:	Registration Page

Description:	Users shall be able to create an account and choose a payment plan for their subscription.
Precondition:	Users do not have an existing account associated with their email address. If the user wants to purchase the service they must have a credit/debit card on hand.
Assumption:	If the user is on the Registration Page they are interested in the service and will purchase a subscription plan.
Test Steps:	1. Navigate to the Login Page 2. Click on the “Register” button 3. Insert users first and last name 4. Create username 5. Insert email address 6. Create password 7. Click “Next” to select a payment plan 8. Choose a payment plan 9. Enter credit/debit number 10. Enter Expiration Date 11. Enter CVV 12. Click “Finish”
Expected Results:	Now a registered user for Redflix and shall be directed to the Main Page

Test Case 3 (Table 3)

Title:	User Authentication and Authorization in SQL
Description:	Using azure backend cloud servers to authenticate users with correct login and subscription plans.
Precondition:	User’s login information should be saved in the Users database.
Assumption:	Users should have a pre-existing account and subscription plan.

Test Steps:	1. Logs in with their username and password 2. The server verifies the credentials on our azure database cloud server and then returns true or false depending on if the credentials are valid or not. 3. If valid, the server allows the user to access the Redflix home page with their subscription plan. 4. If invalid, the server redirects the user back to the login page to retry with their authentication.
Expected Results:	User's with incorrect login will be denied access until correct credentials are entered. User's with correct login, will be authenticated and authorized by the server to access Redflix content.

Test Case 4 (Table 4)

Title:	Media Platform Authentication and Authorization
Description:	When a new user registers with Redflix, the User can add multiple media platform access to retrieve their videos from their preferred media platform.
Precondition:	The user should already have a pre-existing account with their preferred media platform like Youtube, Netflix and Hulu etc.
Assumption:	The user will have different accounts on different media platforms.
Test Steps:	1. When a new user registers an account with Redflix, based on their subscription plan the system will generate an option for them to connect their following media platforms. 2. Once the media platforms have been authenticated, the user can access the videos library from that media platform. 3. If a user fails to authenticate on their following media platforms, an option is given to the user to create a new account for their chosen media platform.

Expected Results:	Once the user has successfully authenticated itself, Redflix adds the media platform account to the user's account and enables adding videos to your library from that particular media platform.

Test Case 5 (Table 5)

Title:	First time user enters sites Homepage
Description:	Upon entering the website, the user shall be presented a grid of all available video media platforms, viewing options, a search bar, a log-in/register button, and any additional content available for the future user.
Precondition:	User is not logged in.
Assumption:	The user is interested to see what Redflix offers.
Test Steps:	1. Should navigate on the Homepage through a web browser. 2. Website shows the different media platforms supported to see if the user wants to purchase Redflix services. 3. Once a purchasing decision is made they will either leave the site or register for the service through pressing the registration button
Expected Results:	The user navigates to the register page via the log-in/register button.

Test Case 6 (Table 6)

Title:	Provide a list of videos.
Description:	User selects the service and the video type that they want and a list of videos will be provided.

Precondition:	The user already has an account with Redflix and is logged into our service
Assumption:	The user wants to select/view the list of videos that they are interested in.
Test Steps:	1. Select the Service button and click the service. 2. Select the Category button and click the type/genre. 3. Connect to the server and send a request. 4. Provide the user with the list of content.
Expected Results:	The user will have a list of video options to select from.

Test Case 7 (Table 7)

Title:	Displaying the 6 videos on the mini screens.
Description:	The user selects videos from the given list and the choice to select upto 6 videos to project it to the mini screens. 1 of the screens will be the big viewable one that the user will mainly be watching. The other 5 will be mini screens surrounding the big main screen.
Precondition:	The user already has a list of videos to select from.
Assumption:	The user wants to watch up to 6 videos at once.
Test Steps:	1. Select the list of videos one by one. 2. Display each selected video to the video player beside it. 3. Select and display upto 6 videos without any interruptions.
Expected Results:	The user can view up to 6 videos in the miniscreen.

Test Case 8 (Table 8)

Title:	Video Functionality
Description:	Users shall be able to fast forward, rewind, pause, adjust volume, adjust video quality, and change subtitle and language options.
Precondition:	User must have selected a video for play
Assumption:	User wants to fast forward, rewind, pause, adjust volume, adjust video quality, and change subtitle and language options to his/her preference.
Test Steps:	1. Navigate the Collection Screen Platform and watch a video selected 2. Video shall show options that will make there video viewing experience more enjoyable 3. Options to click pause, rewind, fast forward, enhance video quality, may be selected 4. Clicking on the speaker icon to adjust the volume via a slide bar. 5. Clicking on the subtitles icon to turn on the subtitles to the languages available.
Expected Results:	When moving 10 seconds forward or backward, a small icon will display in the center of the screen and then fade out after one second. When the user hits pause or play, both buttons icons will display in the center of the screen and fade out after 1.5 seconds. An adjustable slide-bar with a speaker icon at the bottom of the viewing window will adjust the volume. An expandable gear icon will contain the settings for video quality, subtitles, and language options.

Test Case 9 (Table 9)

Title:	Transitioning from selection screen to main screen
Description:	The user transitions from the selection screen to the main screen.

Precondition:	The user has selected upto 6 videos to watch.
Assumption:	The user wants to watch the 6 videos in a full screen mode.
Test Steps:	1. Select videos from the list and have up to 6 videos on the selection screen. 2. Select one of the videos from the 6 screens to emphasize it. 3. Transition to the main viewing page where the selected video from the selection screen will be emphasized.
Expected Results:	The user is in the main viewing screen where only the videos will be displayed and the user selected video in the selection screen will be emphasized.

Test Case 10 (Table 10)

Title:	Adding videos to library
Description:	The user can save the videos that they like to the library where they can watch it later. Each video will have a star in the top left corner, when clicked it will turn red and become one of the users favorites page
Precondition:	The user has already logged in and has a list of videos to select.
Assumption:	The user wants to save the videos to view for later.
Test Steps:	1. Select the star option to the next of each listed videos in the selection screen. 2. After selecting the option, the videos will be added to the library page. 3. Select the library option in the selection screen to go the library page.

Expected Results:	The user will have list of videos in the library section

TRACEABILITY OF TEST CASES TO USE CASES:

(Table 11)

Test Case / Use Case	1	2	3	4	5	6	7	8	9	10
1	X	X								
2						X	X			
3										X
4								X		
5									X	
6			X							
7				X						
8					X					

TECHNIQUE FOR TEST GENERATION:

Test cases are designed based on the functionalities of an application. So it varies from one application to another. The purpose of test case generation is to check the output against expected results. Based on the results, either the test case is modified or kept as it is.

The aim of testing is to find bugs in a system or application. Test case generation is the process of building test suites for detecting system errors. A test suite is a group of relevant

test cases bundled together. Test case generation is the most important and fundamental process of software testing.

There are multiple techniques available for generating test cases:

- Goal-oriented approach – The purpose of the goal-oriented test case generation approach is to cover a particular section, statement or function. Here the execution path is not important, but testing the goal is the primary objective.
- Random approach – The random approach generates test cases based on assumptions of errors and system faults.
- Specification-based technique – This model generates test cases based on the formal requirement specifications.
- Source-code-based technique – The source-code-based case generation approach follows a control flow path to be tested, and the test cases are generated accordingly. It tests the execution paths.
- Sketch-diagram-based approach – This type of case generation approach follows the Unified Modeling Language (UML) diagram to formulate the test cases.

Functional Testing :

Acceptance Testing - An Acceptance Test is performed by the client and verifies whether the end to end the flow of the system is as per the business requirements or not and if it is as per the needs of the end-user. Client accepts the software only when all the features and functionalities work as expected. It is the last phase of the testing, after which the software goes into production. This is also called User Acceptance Testing (UAT).

Beta Testing - Beta Testing is carried out to ensure that there are no major failures in the software or product and it satisfies the business requirements from an end-user perspective. Beta Testing is successful when the customer accepts the software. Usually, this testing is typically done by end-users or others. It is the final testing done before releasing an application for commercial purpose. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area. So the end-user actually uses the software and shares the feedback to the company. Company then takes necessary action before releasing the software to the worldwide.

Back-End Testing - In Back-end Testing GUI is not involved, testers are directly connected to the database with proper access and testers can easily verify data by running a few queries on the database. There can be issues identified like data loss, deadlock, data

corruption etc during this back-end testing and these issues are critical to fixing before the system goes live into the production environment

GUI Testing - The objective of this GUI Testing is to validate the GUI as per the business requirement. The expected GUI of the application is mentioned in the Detailed Design Document and GUI mockup screens. The GUI Testing includes the size of the buttons and input field present on the screen, alignment of all text, tables, and content in the tables. It also validates the menu of the application, after selecting different menu and menu items, it validates that the page does not fluctuate and the alignment remains the same after hovering the mouse on the menu or sub-menu.

Non Functional Testing :

Risk Based Testing - In Risk-Based Testing, the functionalities or requirements are tested based on their priority. Risk-Based Testing includes testing of highly critical functionality, which has the highest impact on business and in which the probability of failure is very high. The priority decision is based on the business need, so once priority is set for all functionalities then high priority functionality or test cases are executed first followed by medium and then low priority functionalities. The low priority functionality may be tested or not tested based on the available time.

Static Testing - Static Testing is a type of testing which is executed without any code. The execution is performed on the documentation during the testing phase. It involves reviews, walkthrough, and inspection of the deliverables of the project. Static Testing does not execute the code instead of the code syntax, naming conventions are checked. Static Testing is also applicable for test cases, test plan, design document. It is necessary to perform static testing by the testing team as the defects identified during this type of testing are cost-effective from the project perspective.

Security Testing - Security Testing is done to check how the software or application or website is secure from internal and external threats. This testing includes how much software is secure from the malicious program, viruses and how secure and strong the authorization and authentication processes are. It also checks how software behaves for any hackers attack and malicious programs and how software is maintained for data security after such a hacker attack.

EVIDENCE THE TEST CASES, DOCUMENT HAVE BEEN PLACED UNDER CONFIGURATION MANAGEMENT:

REFERENCES:

1.1. https://www.ieee.org/about/corporate/governance/p7-8.html
1.2. https://ethics.acm.org/code-of-ethics/software-engineering-code/
1.3. https://www.netflix.com/
1.4. https://www.hulu.com/
1.5. https://www.youtube.com/premium
1.6. K. Pogrebnoy, “How to Create a Video Streaming Website like Netflix, Amazon, or Hulu,” *CodeTiburon*, 31-Oct-2019. [Online]. Available: https://codetiburon.com/create-video-streaming-website-like-netflix-amazon-hulu/. [Accessed: 14-Feb-2020].