Software Requirements Document

for

Open-Source Real-Time Video Player

Version 4 approved

Prepared by Brian Hernandez, Mirasol Davila, Wendy Joya, Ashley Jetty, Israel Lopez-Diaz, Tim Ellis, Rafael Mendoza, Jeffrey Luu, David Melendez

AT&T

May 14, 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First Draft #1	12-06-2020	Created	1
Second Draft v1	03-04-2021	Updating progress of the project	2
Second Draft v2	04-21-2021	Updating	2.1
Third Draft	05-01-2021	Updating for final approval	3
Final Draft	05-14-2021	Final revision	4

1. Introduction

1.1 Purpose

The purpose of the Software Requirements Documentation (SRD) is to outline the requirements of our Open-Source Real-Time Video Player. We will illustrate the purpose, external interface requirements and nonfunctional requirements of the project. We will specify the functionalities throughout this document.

1.2 Intended Audience and Reading Suggestions

This documentation is intended for developers and users. It is broken down into its software components where it will make it easier to understand our project. This document contains the overall idea of how our different components work together. A suggestion for users will be to look into the sections of understanding the purpose, the overall description and acronyms to have a sense of the project.

1.3 Product Scope

Our goal is to provide additional functionalities to our Open-Source Video player to provide an operational web-app with the functions of automated testing, recording of data to an excel sheet, and network throttling to simulate various testing conditions.

Objective:

- Allow users to input:
    - JSON file with list of video urls
    - JSON file with list of network profiles
- Display and record the following KPI metrics:
    - Video Start Time
    - Rebuffering Ratio
    - Average Bitrate
    - Bandwidth
- Automate the following process:
    - Read and apply one video url, on the video player, one at a time
    - Apply each network profile on each video
        - Read the network profile file and apply each value every second
    - Keep track of all the data
    - Once all videos are finished playing, download an xlsx file with all the data locally
- Output xslx data should include:
    - Tab containing video start time data

- Tab containing rebuffering data
- Tab containing bitrate data
- Tab containing network data
- Tab containing other important metrics such as frames per second, video resolution, etc

1.4 Definitions, Acronyms, and Abbreviations

ORVP	Open-Source Real-Time Video Player
SRS	Software Requirement Specification
DASH	Dynamic Adaptive Streaming
HLS	HTTP Live Streaming
KPI	Key Performance Indicator
VST	Video Start Time
RBF	Rebuffering
ABR	Adaptive Bitrate
NTWK	Network
VMAF	Video Multimethod Assessment Fusion
Broadband	Telecommunication for Data transmission
REPOS	Repositories
JavaScript	An interpreted language which is not compiled
SS	Stream Saver
Selenium	Automation for web applications
QA	Quality Assurance
Bandwidth	The maximum amount of data transmitted over an internet connection in a given amount of time
UI	User Interface

1.5 References

- HLS.js: https://github.com/video-dev/hls.js/
- HLS Demo: https://github.com/video-dev/hls.js/tree/master/demo
- Selenium: https://www.selenium.dev
- Chart.js: chartjs.org
- Mozilla MDN: https://developer.mozilla.org/en-US/
- MyExcel: https://github.com/jsegarra1971/MyExcel
- Node.js https://nodejs.org/en/
- Javascript: https://www.javascript.com

2. Overall Description

2.1 System Analysis

This software system runs independently of any other software system. To collect data using this system, we store it locally on the owner's computer. It's done by placing it in a xlsx file. Additionally some of the other data and metrics are displayed in the UI for real time analysis. Selenium was used to control all areas from our project, including reading a list of video urls and sending them to our video player, reading a list of network profiles and applying them every second using the chrome dev tools, keeping track of all the data, and lastly to output an xlsx file containing all the data, locally.

Technical hurdles:

1. Location of saved data and retrieval of data
2. Broadband speed must be faster than the network profiles applied

Solution for technical hurdles:

1. At load-time, we should have a database that will hold all the data collected from each user with security measures to keep the data private.
2. Using our webpage, we will have a reference of what kind of speed it’s supporting and best when loading videos.

2.2 Product Perspective

The product will be a webpage for a video player with multiple functionalities which shall assist the user with qualitative analysis. Multiple versions of the product may be implemented for each individual video player.

2.3 Product Functions

Major functions in the software to provide efficiency to our users include:

- User input
    - List of video urls
    - List of custom network profiles
- Configurations adjustable on the video player
- KPI metrics implemented to ensure accuracy and best capabilities such as:
    - VST
    - Rebuffering Ratio
    - Video Quality
    - Average Bitrate
- Real-Time network throttling
- Selenium
    - Read from list of video urls and have our video player play them one at a time
    - Read values from the network profiles and apply them every second using the chrome dev tools
- Spreadsheet collection done locally

The functions combined together help illustrate the relationships needed for the whole software. Our goal for this software is to provide a high quality video player with minimal rebuffering time plus full advantage of the functionalities.

2.4 User Classes and Characteristics

Users Classified:

- Developers:
    - California State University, Los Angeles Senior Design Team
- Users:
    - Users within the AT&T company
    - General Public

Our primary demographics will be developers because this software is intended to be used as a testing environment. Where videos are runned to see how fast they perform with fulfillment of the functionalities implemented in the software.

2.5 Operating Environment

The environment where the software will live would be locally on the users computer internet connection. Any operating system is compatible with this software which is beneficial for completing

tests with this software. Hardwares specifications of the user will help understand the differences in datasets. Differences in hardware, operating systems, or system software will not impact a user's ability to utilize this software.

2.6 Design and Implementation Constraints

Like any software design there are always constraints that are put in place for the protection of its software and users. There are some constraints placed in this software which are:

- Repos were provided by AT&T
- IP Addresses were granted access with the help of the AT&T team (Pv4)
- Data Constraints
- Data Caps
- HLS Videos
    - Video players are HLS supported
        - HLS.js player is only HLS video supported
- Security Considerations
    - Data collected should only be used for testing purposes
- Internet Connectivity/Broadband
- Web browser:
    - Google Chrome

Expansion of these constraints will be mentioned below under non-functional requirements

2.7 User Documentation

TBD

2.8 Assumptions and Dependencies

Requirements:

- Video Access
- Broadband
    - Network Profiles: WIFI, Cellular, and Stream Saver
    - IP Address

The main concern would be broadband because without internet connection the webpage wouldn't be able to display the html file. It's speed would affect the buffering ratio and video quality as well.

2.9 Apportioning of Requirements

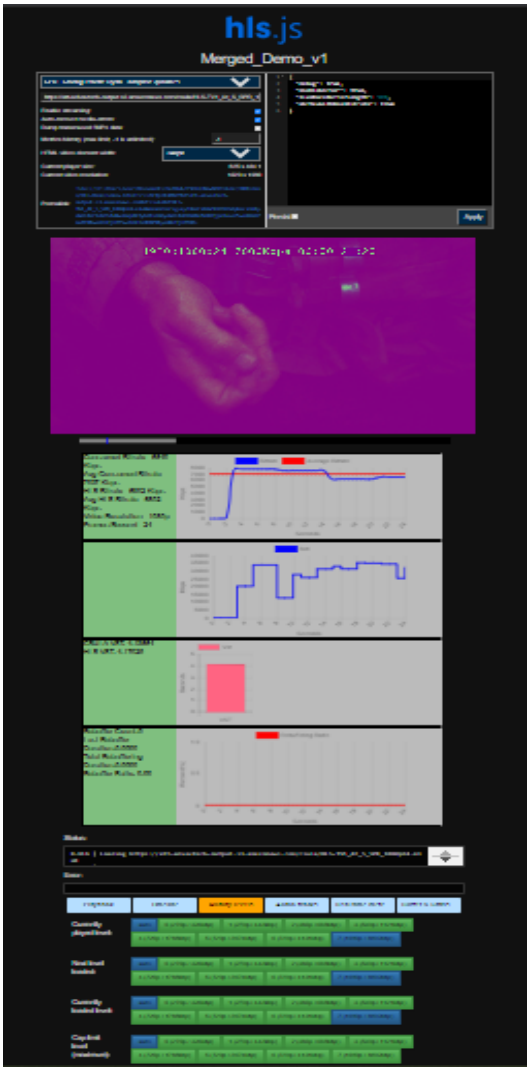
Requirements that will be delayed until a future version of the software:

- VMAF and/or SSIM
- Database to store all users data collected
    - Database such as a Server
    - Updated UI to be a user-friendly UI
- DASH videos to be supported by our video player
- Use the Shaka player and extend it with the same functionalities we added to the Hls.js player
- Video Player Performance Analysis #2
- Website Hosting

3. External Interface Requirements

3.1 User Interfaces

When the user opens the application on a Chrome browser, they will see the entirety of the NVKE and its UI components. Visually going from top to bottom, the user will first see the Video Source Information which provides title and source information of the video selected. Beneath the Video Source Information is the Video Player Screen which is the viewer that plays the video and the KPI Metric Stats that are values of the real time metrics of the video being played. Underneath the Video Player Screen are the KPI real time charts. These allow the user to view the metrics changing in real time. Lastly, the HLS-DEMO's real time metrics.



3.2 Hardware Interfaces

Hardware Required:

- The device’s resolution must support the video’s max resolution to properly stream the videos
- Device used must be a laptop or desktop computer
- It is preferred to have a cpu and gpu with good performance
- It is preferred to have fast broadband speeds available for best results when applying network profiles

3.3 Software Interfaces

A software requirement that’s needed to be able to use the ORVP application would be:

- Mac, Window, or Linux
- Broadband Connection
    - WIFI
    - Network Throttling
- For testing purposes: Selenium
- Web browser requirements
    - Google Chrome - Version 90.0.4430.212 or newer

3.4 Communications Interfaces

Communication functions required in association with the software would mainly be web browsers. For a future version of this software, a network server communications protocol should be implemented. This will be helpful to retrieve data, where at the moment we are storing it locally. Having a communications protocol would essentially make it easier to analyze data collected from each user. After doing so, there's a matter of security that is now involved to maintain data and user information confidentially.

4. Requirements Specification

4.1 Functional Requirements

Guidelines for functionality:

1. The software shall take a pre-set json file with a list of video urls by the user.
2. The software shall take a pre-set json file with a list of network profiles by the user.
3. The software shall perform analysis by recording data on the video playback.
4. The software shall locally download an xlsx file containing all the data recorded.
5. The software shall be able to play HLS videos.
6. The software shall display the different KPI metrics while running the web page.
7. The software shall plot the video start time.
8. The software shall display the buffering graphs.
9. The software shall display the average bitrate graphs.
10. The software shall display the bandwidth and network profile graphs.
11. The software shall handle errors by returning warnings with invalid data.
12. The software shall be able to maintain the video that was previously loaded even if the web browser was reloaded.

4.2 External Interface Requirements

Input Requirements:

- Network Profiles: Excel files that users can provide. The Network Profiles consist of different download speed values to apply to the Network Video KPI Emulator. The following Network Profiles are provided.
    - WIFI Data
    - Cellular Data
    - Stream Saver
- profiles: A json file where the user can append or remove network profiles they want to apply.
- playlistdata: A Javascript file is the file where the user can append or remove video urls.
- playlist: A json file where the user determines what videos to play from the playlistdata JavaScript file.

Output Requirements:

- Workbook file that contains five excel worksheets
- The following worksheets are:
    - SUMMARY
    - VST

    - RBF
    - ABR
    - NTWK
- Data: such as the KPI metrics

The purpose of the inputs required is so that the software has some data to use and look for while running. For the output requirements the data was written into CSV files and the data collected is also put into the CSV file.

4.3 Logical Database Requirements

Does not apply.

4.4 Design Constraints

The software has some constraints concerning the type of videos being imported on to the video players. This has to do with the fact that not both players can support HLS videos. The design for this software was with the help of the AT&T team granting us access to their repos and IP addresses. Other design constraints consist of data caps and data constraints. This is because internet speed always varies from user to user. Internet traffic might be heavy for one and the other could have less traffic which also depends on time zones and locations. Lastly, security takes a big part in our design because data could be used apart from just being for testing purposes.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Web application requires you to run on Chrome.
- Web application requires the web browser to support HTML 5.
- To have access to all the functions in our web application, it is required to run on a desktop or laptop.
- Internet Connection is required
- To get the best results, fast and stable bandwidth is preferred
- Many users can be on our web-app simultaneously
- Since our software is automated using selenium and consists of opening and closing applications, it is preferred to have a cpu and gpu with good performance

5.2 Safety Requirements

A safety requirement if open to the general public there should be restrictions to protect the developer and user. Making the software unable to be tampered with for the security and integrity of the user's information and data.

5.3 Security Requirements

Our software requires security for protection and to improve time efficiency. Security measures can protect the security and integrity of the project. Security measures involved are as follows:

- Data collected should not be:
    - Forged
    - Replicated
    - and/or Sold
- Video replication:
    - Videos provided for testing may not be replicated or shared in any form without tacit consent as it may violate their integrity
- Servers:
    - Information stored on the server can only be accessed by a project leader and/or company's permission.
    - Access only thru IP Address whitelisting

5.4 Software Quality Attributes

This software offers the highest level of quality in order to ensure flexibility, reliability, correctness, test-ability, and usability. For example, we are ensuring the accuracy of our KPI metrics.

Multiple tests are performed to ensure accurate KPI data. The interface is made to be as user friendly as possible by making it easier to view every KPI implemented. To ensure the quality of the software, multiple tests are performed in the quality assurance process.

5.5 Business Rules

The functionalities added on top of the open source video players currently belong to AT&T and California State University, Los Angeles.

6. Legal and Ethical Considerations

This section covers ethical and legal considerations that must be considered while developing, testing, delivering, and publishing the Open-Source Real-Time Video Player (ORVP). Every one of our team members must keep in mind the ACM code of ethics when working on and with the ORVP project, however we have highlighted the most crucial sections applicable to the ORVP. The topics covered are the following: Data Collection Policies, Streaming Copyrighted Content, Security and Reliability.

ORVP collects data to operate effectively and provides the Host (As of 12/11/2020, AT&T) with information that will be analyzed to help improve their streaming services and platforms. Users provide some of this data directly, such as when the user asks for access to the ORVP application, your IP Address is used to grant you access for video playback. We obtain USER-AGENT information which includes information about a user's browser and the device requesting the ORVP application. Finally, Information is being recorded when a user interacts with this application: such data will be given to the Host and is crucial for the ORVP application, whose goal is to provide Clients with improved services. We are aware that mishandling data can result in unwanted information being published online, therefore whoever hosts this application must disclose how they handle the data and provide its user the confidence to use the application. The Host shall be responsible for any data collected, and compliant with any, county, city, state, or country laws. Standalone, the ORVP application does not collect Gender, Age, Social Security Numbers, Home Addresses, or any data that can be linked to a specific person.

When streaming or distributing any type of content we must take into consideration the laws and protections that copyrighted content has. Motion content does not need to include a copyright notice or be registered with the U.S. Copyright Office and receive copyright protection. Our application standalone does not contain any Videos, but however it links to content provided by the Host of the applications. As soon as a Host adds a link to any digital medium, they are responsible for complying with the copyright protection laws. ORVP is not responsible for how a Host may or may not abuse such laws. A host must comply with the laws set by the U.S. Copyright Office, and any future laws regarding the distribution of video content over the world wide web. All material used must have licensed approval of usage as required by law.

Security is a big influencer when developing any web application. Although the project was built on open source code and tested by third party groups and organizations, we must consider cyber security and take preventative measures for cyber-attacks. All source code should be revised before deployment, by the developers and the deployment teams. It is our responsibility that the application must do what it is documented to do and nothing more. Any type of malicious cyber-attack must be considered and prevented to our best ability. When working with open source material it is much harder to find responsible parties if errors arise but nonetheless it is crucial to deploy well tested and secure applications.

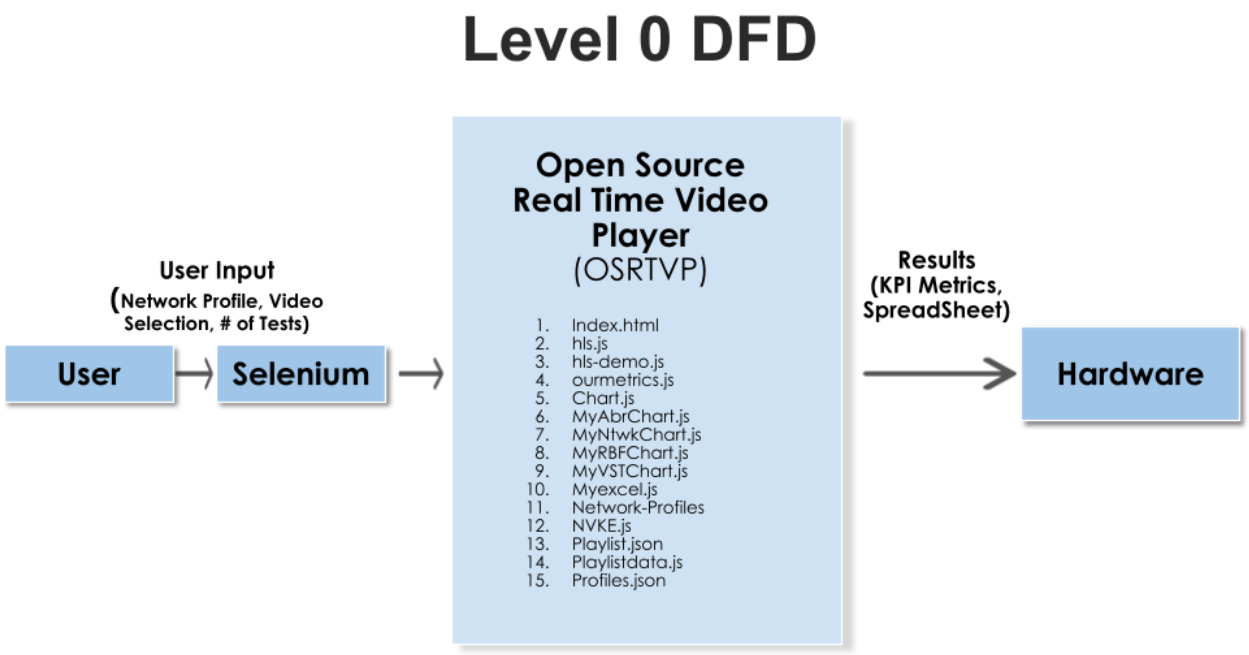
Finally, Reliability is kept in mind, we will not promise anything that can't be accomplished or done with the help of this application. The ORVP is intended to help build and strengthen streaming

services by providing real-time metrics and data based on simulated network conditions. The team is aware and is responsible for providing data results, reports and explanations. In the case that our current host, AT&T, implements this project to their applications or publishes this project as Open Source, the hosts are responsible for analyzing the data and running their own set of tests to assure reliability and accuracy. We will provide some metrics, and data accuracies, but they are to be analyzed and recalculated if needed.

Appendix A: Glossary

Reference Section 1.4

Appendix B: Analysis Models



Appendix C: To Be Determined List

1. User Documentation
2. Logical Database Requirements