

Software Requirements Specification

for

Open-Source Real-Time Video Player

Version 1 approved

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

AT&T

December 6, 2020

Table of Contents

Table of Contents.....	02
Revision History.....	04
1. Introduction.....	05
1.1. Purpose.....	05
1.2. Intended Audience and Reading Suggestions.....	05
1.3. Product Scope.....	05
1.4. Definitions, Acronyms, and Abbreviations	06
1.5. References.....	06
2. Overall Description.....	07
2.1. System Analysis.....	07
2.2. Product Perspective.....	07
2.3. Product Functions.....	07
2.4. User Classes and Characteristics.....	08
2.5. Operating Environment.....	08
2.6. Design and Implementation Constraints.....	08
2.7. User Documentation.....	09
2.8. Assumptions and Dependencies.....	09
2.9. Apportioning of Requirements.....	09
3. External Interface Requirements.....	10
3.1. User Interfaces.....	10
3.2. Hardware Interfaces.....	10

3.3. Software Interfaces.....	11
3.4. Communications Interfaces.....	11
4. Requirements Specification.....	12
4.1. Functional Requirements.....	12
4.2. External Interface Requirements.....	12
4.3. Logical Database Requirements.....	12
4.4. Design Constraints.....	13
5. Other Nonfunctional Requirements.....	14
5.1. Performance Requirements.....	14
5.2. Safety Requirements.....	14
5.3. Security Requirements.....	14
5.4. Software Quality Attributes.....	14
5.5. Business Rules.....	15
6. Legal and Ethical Considerations.....	16
Appendix A: Glossary.....	18
Appendix B: Analysis Models.....	18
Appendix C: To Be Determined List.....	18

Revision History

Name	Date	Reason For Changes	Version
First Draft	12-06-2020		1

1. Introduction

1.1 Purpose

The Software Requirements Specifications (SRS) document dives into our projects' software specifications. Where it will expand the knowledge of whomever reads this document to serve its purpose to help understand the specifications. We will illustrate the purpose, external interface requirements and nonfunctional requirements of the project.

The purpose of this document is to specify the requirements of ORVP. Specifically, the requirements include the video players, the software used to create the players, libraries and metrics implemented. We will specify the functionalities throughout this document.

1.2 Intended Audience and Reading Suggestions

This documentation is intended for developers and users. It's 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

In this section, our goal for our project is to provide users a very functional video player that supports both HLS and DASH videos, and extended functionalities.

- Objective:
 - Maintain maintenance of how the video player keeps performing everytime runned
 - Run selected videos with the extended functionalities implemented
 - Keep the videos running at a high speed, depending on network (broadband)
 - Keep the videos running at a high speed (broadband)
 - KPI reporting capabilities
 - Display VST, rebuffering Ratio, and VMAF

- Simulate a real-time network throttling

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
VMAF	Video Multimethod Assessment Fusion
Broadband	Telecommunication for Data transmission
REPOS	Repositories
JavaScript	An interpreting language not a compiler

1.5 References

- ExoPlayer: <https://github.com/rc728m/ExoPlayer.git>
- Exoplayer Demo: <https://exoplayer.dev/demo-application.html>
- Shaka: <https://github.com/rc728m/shaka-player.git>
- HLS.js: <https://github.com/rc728m/hls.js>
- HLS Demo: <https://hls-js-dev.netlify.app/demo>
- Video.js: <https://github.com/rc728m/video.js.git>
- Clappr Stats: <https://github.com/clappr/clappr-stats>
- Mux.js: <https://github.com/videojs/mux.js#readme>
- Chart.js: chartjs.org
- MyExcel: <https://github.com/jsegarra1971/MyExcel>

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 an csv file, plus some of the other data is seen on the display window. The goal for this project is to implement a video player that can have accessible functionality to all our users.

Technical hurdles:

1. Location of saved data and retrieval of data
2. Broadband speed

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
- Selection of video to run in the video player are pre-defined in the software
- Configurations
- KPI metrics implemented to ensure accuracy and best capabilities such as:
 - VST
 - Rebuffering Ratio
 - Video Quality
- Real-Time network throttling
- 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:
- Users:
 - Users within the company, AT&T, etc.
 - 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 beneficiary if completing tests with this software. Different hardwares will help understand the different datasets in correlation with the different operating systems. Having different hardware platforms, operating systems and different softwares will not be affected by our 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
- Data Constraints
- Data Caps

- HLS Videos
 - Video players are HLS and DASH supported
 - HLS.js player is only HLS video supported
- Security Considerations
 - Data collected should only be used for testing purposes

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

2.7 User Documentation

TBD

2.8 Assumptions and Dependencies

Requirements:

- Broadband
 - WIFI Speed
 - 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

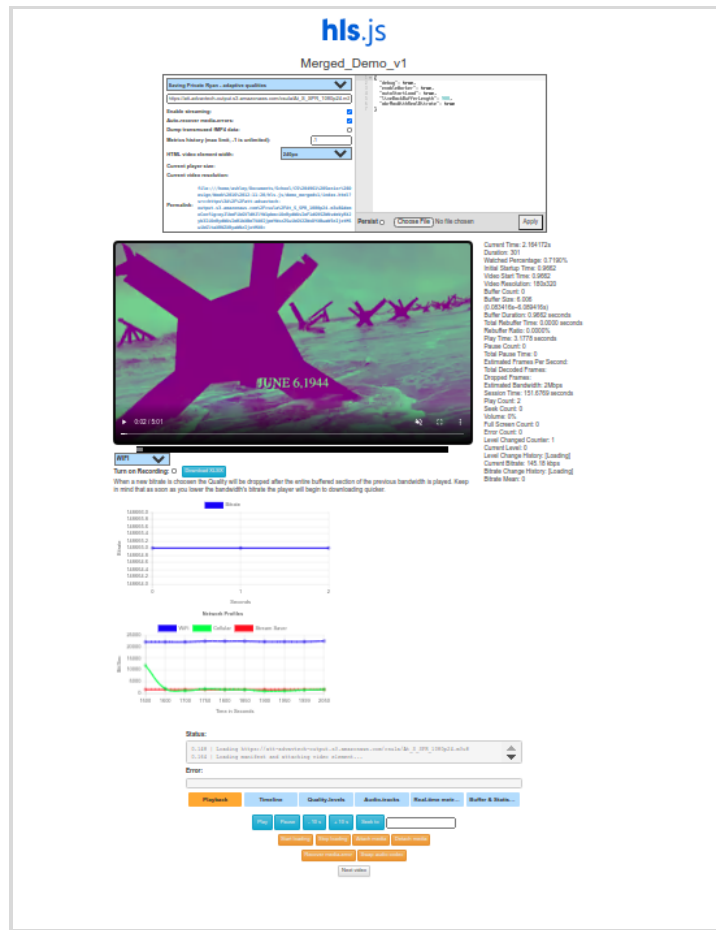
Requirements that will be delayed until a future version of the software:

- VMAF
- Database to store all users data collected
 - Database such as a Server

3. External Interface Requirements

3.1 User Interfaces

Application will start up on a Google Chrome or Safari browser. Then they will see the General Overview of the ORVP. With the location of the metrics and KPI's on the right side of the screen. At the top, the name of the Video player can be observed and if you look underneath the name there's a toggle box where you can find the different videos already pre-defined. The center of the screen is where the video chosen will be displayed. At the bottom, data concerning bitrate shown in a real time linear graph where it's modifying as the video plays and network profiles. If an error occurs then an error message will display.



3.2 Hardware Interfaces

Hardware Required:

- Support at least 1080p

- For playing HD

3.3 Software Interfaces

A software requirement that's needed to be able to use this document would be:

- Web browser requirements
 - Compatible browsers
 - Google Chrome and Safari
 - Browser needs to be up-to-date

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 amount of videos to be chosen by the user.
2. The software shall perform analysis by recording its data.
3. The software may have an alternate button for alternating between HLS.js or Shaka video player.
4. The software may be able to play both HLS and DASH videos even if they aren't supported by the player.
5. The software shall display the different KPI metrics while running the web page.
6. The software shall display the buffering graphs when video is playing.
7. The software should handle errors by returning invalid data.
8. The software should 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
- WIFI

Output Requirements:

- CSV files
- 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

TBD

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 both HLS and DASH 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 Safari or 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.

5.2 Safety Requirements

N/A

5.3 Security Requirements

Our software requires security to make our software more protected but at the same time efficient. Security reasons that can have repercussions can affect the integrity of the project and software. Therefore placing some security requirements can help us prevent:

- Data collected shouldn't be:
 - Forged
 - Replicated
 - and/or Sold
- Video replication:
 - Can't be replicated without the consent of the project leader and/or company.
 - Can contaminate the integrity of the video
- Servers:
 - Information stored on the server can only be accessed with project leader and/or company's permission.
 - Access only thru IP Addresses

5.4 Software Quality Attributes

This software is taking in consideration the highest type of quality where it will ensure flexibility, reliability, correctness, test-ability, and usability. For example, we are ensuring the accuracy of our KPI metrics. We are making multiple tests to ensure that we are getting our best video quality, rebuffering ratio and etc. When it comes to its interface we tried making it as user friendly as possible. Where it's easier to maneuver around the web page with every KPI implemented to be displayed.

5.5 Business Rules

The functionalities added on top of the open-source real-time video players are reserved under the rights of AT&T.

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 for the ORVP. The topics covered will be touched are the following: Data Collection Policies, Streaming Copyrighted Content, Security and Reliability.

ORVP collects data to operate effectively and provide the Host (As of 12/11/2020, AT&T) with information that will be analyzed to help improve their streaming services and platforms. You 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, which 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 Names, Gender, Age, Social Security Numbers, Addresses, ISP data, 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 to 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. If a host does not take Copyright protection laws, they are in direct violation of the law and can be fined by the Federal Bureau of Investigation if accused and found guilty of white-collar crimes. 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.

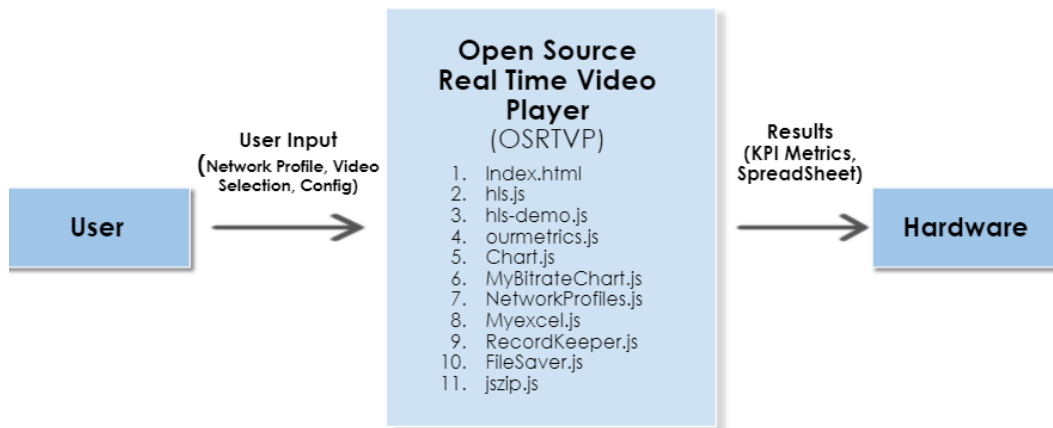
Finally, Reliability is kept in mind, we will not promise anything that can't be accomplished or done with the help of this application. The ORVP is intended to help build and strengthen streaming services by providing real-time metrics and data based on simulated network conditions, created by modifying HLS.js and Shaka Player, open source video players. 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

Level 0 DFD



Appendix C: To Be Determined List

1. User Documentation
2. Logical Database Requirements
3. Safety Requirements