Capstone Project: “Find That Show” SRS

Prepared by Kahle Broadnax

Central Washington University

1/30/2022

CS 481

Table Of Contents

1. Introduction:

The following document is meant to serve as the Software requirements specification for an online search engine software (website) meant to enable the efficient search of television shows and movies across a range of all current streaming services via web scraping.

1.1 Purpose:

This SRS is intended both to be used by the team working on it to ensure progress is being made, and that the requirements laid out here are being followed, but also as supplementary material for two presentations during the course of development, for which this document is a part of them and required.

1.2 Scope:

The product being developed is intended to allow a user to easily search for television shows and movies across all the current streaming services by utilizing web scraping. The additional features included to make this more robust than other search engines would be search parameters you can enable and disable, including specific streaming services, prices of those subscriptions, and the regional availability of shows and movies. Additionally, there is an intent to allow the user to search by genre across the streaming services, instead of requiring an exact show name. This document will also contain some aspects of a design document, which will be expanded upon in the future, but for now, will include the High-level and Low-Level design.

1.3 Intended Audience:

The intended audience is both the developers of the product, which need an SRS for documentation of features and design goals, and the professor of the CS 481 class at Central Washington University, Professor Razvan Andonie.

1.4 Intended Use:

All members of the development team are to be given access to this document for evaluation regarding its contents. It is intended to make their workflow easier by laying out the requirements for the software and to identify what features are critical and which are non-essential but preferred (Functional and Non-functional requirements).

1.5 Definitions and Acronyms:

Below is a running list of definitions and acronyms. More may be added as the document expands:

SRS: Software requirements specifications
CWU: Central Washington University
Q&A: Questions and answers
Web scraper: A program that parses websites for relevant information depending on the programmer's needs
VPN: Virtual Private Network. Allows for users to reassign their web-based location, essentially giving access to content outside of their region
Dub: A vocal track intended to replace the original dialogue of a show or movie. Typically, this is in a different language than the original to give non-native speakers an opportunity to understand a show or movie without resorting to subtitles.

2. Overall Description of Software:

This section will contain a description of the software, followed by a series of questions and answers regarding the specifics of the software to consider for the formal requirements. The Q&A section will cover the user needs, assumptions, and dependencies.

2.1 Description of Software:

The software to be developed is a website that allows for the convenient search of any television show or movie across a range of current streaming services, both free and paid. The GUI frontend would allow for the selection of specific filters, such as by streaming service, by price, by region, or by genre. The search aspect of it is achieved via a backend web scrape search. The main backend feature that is intended to be implemented is regional searching, so if a show was available in the UK for instance, but the user was in the US, the user would be made aware of that availability.

2.2 Questions and Answers:

Question: Who is the software intended for?
Answer: The software is intended for users with access to multiple steaming services and or users with access to a VPN.

Question: What are the coding languages the software will be developed in?
Answer: Currently, the front end is written in javascript with a React framework. The backend is written in Python, Beautiful Soup, and SQLite. GitHub is being used to document commits and project updates.

Question: What external libraries are to be considered?
Answer: Any free or open-source library is available to use. As long as the implementation consists of more than calling a library function and using that as the entire project, it is able to be used.

Question: What is the timeframe for this project?
Answer: This project has a timeframe of 9 weeks, the length of the 2022 winter quarter of CWU. A fully implemented project is expected by the end of the timeframe, so features will be dropped as necessary to ensure a functional project by the end of the 9 weeks.

Question: What potential obstacles are there in the development of this software?
Answer: The software developers of this project are less experienced in Python and javascript, and the developers have other classes to work on as well. Managing the time spent on the project, in meetings, and learning the languages more fully are the main obstacles to this project.

Question: What constraints does the development face?
Answer: No budget is available, only free or open-source materials can be used.

Question: Given this service is intended for VPN integration and support, what languages are supported?
Answer: Subtitles and dubs for all languages available on a given streaming service is intended to be supported. The website itself will be presented and written in English.

Question: Does the website intend to use a user account system to keep track of user preferences and previous search queries/options?
Answer: If time is permitting, user accounts will be created and integrated into the software. Currently, cookies are intended to perform a similar functionally if they can be implemented, else no previous search query functionally will be implemented.

3.1 Requirments:

This section will lay out the functional and non-functional requirements, as well as an analysis of them.

Requirement #1: The software will accept typed user input pertaining to what show they want to search for

Analysis: This is a functional requirement
Validity: This is congruent with the software description
Feasibility: Accepting user input is easy to implement
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is mandatory for general software functionally

Requirement #2: The software will search for a given show the user has inputted using web scraping

Analysis: This is a functional requirement
Validity: This is congruent with the software description
Feasibility: Web scraping is a well-documented functionally
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is mandatory for general software functionally

Requirement #3: The software will be designed for personal computers as a website

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description
Feasibility: Designing websites is well-documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is mandatory for general software functionally, but the project could expand or shift to mobile development if more time was permitted or encouraged

Requirement #4: The software should utilize servers (if applicable)

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description

Feasibility: Utilizing servers to host websites is well documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, and should be worked on after a local website is shown to function. Since this software has no budget, it is not a priority at the moment

Requirement #5: Cookie use must be agreed to by the user

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description
Feasibility: Cookies are well documented
Consistency: This doesn't interfere with other tasks the software will perform and is more or less a requirement for most websites
Prioritization: This is not mandatory for the software to function, but could improve the user experience by remembering preferences. Accounts could perform the same functionally, but those have less priority than cookies at the moment

Requirement #6: The software should have the ability to access external websites to direct the user to

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description
Feasibility: Connecting websites together via hyperlinks and links is well documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, and could face some integration issues depending on how extensive the integration is (link directly to the web page the show was found on or just the website)

Requirement #7: Functional testing needs to be performed throughout the creation of the software

Analysis: This is a functional requirement
Validity: This is not incongruent with the software description
Feasibility: Unit tests are well documented, and required to assess the functionality of a software
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, but is encouraged greatly

Requirement #8: The software will allow the user to search by genre or language instead of by show

Analysis: This is a non-functional requirement
Validity: This is congruent with the software description
Feasibility: Creating genre filters to input into the web scraper is documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly for users who want to search for shows they don't know the name of of

Requirement #9: The software will have filter settings to enable and disable specific search options such as subscription price and service

Analysis: This is a non-functional requirement
Validity: This is congruent with the software description
Feasibility: Changing the parameters that go into a web scraper is well-documented
Consistency: This doesn't interfere with other tasks the software will perform, except for limiting search results
Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly for users who want to limit or specify certain streaming services in case they don't own all of them

Requirement #10: The software should be easy to use

Analysis: This is a non-functional requirement
Validity: This is not incongruent with the software description
Feasibility: Making a user-friendly interface is well-documented
Consistency: This doesn't interfere with other tasks the software will perform
Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly

Requirement #11: Results should appear within 20 seconds

Analysis: This is a non-functional requirement
Validity: This is not incongruent with the software description
Feasibility: Tests can be run to determine how to best increase accuracy and speed
Consistency: This doesn't interfere with other tasks the software will perform

Prioritization: This is not mandatory for the software to function, but would increase the user experience significantly. The software should be accurate first, then be fast if one has to take priority over the other