

Software Requirements Specification Version 2.0

February 22, 2019

Modular Multimedia Streaming Platform

Infinity Bits LLC

Owen Dugmore

Alessandro Cois

Sydney Ho

Steve Oseguera

Jorge Avalos

Project Overview

The site will be hosted on a remote server that can be accessed through a web browser by the user. The website will provide various functionality to the user, as described in the functionality section. Most functionality requires that the user creates an account. There is no immediate charge for creating an account. The account allows the user to watch media of their choosing and interact with other users of the site.

Scope of Project

The scope of this project is to create an infrastructure with a modular design that will allow a user to select a desired media from a selected list of producers such as Amazon Prime Video, Netflix, Hulu, ect. The website will be subscription based with a monthly fee that is dependant on the amount as well as type of media that is selected for viewing by the user. The videos will be represented on the users main page for easy access, but the user will also be shown previews of other selected media that is based upon the user's history of viewed videos, popular shows or movies that is based on the amount of total views that the video has, and a favorite genre that is selected by the user. This feature is what will set this streaming service apart from our competitors. However, this is not the only redeeming feature of this service. There will also be a social aspect.

The social aspect will include, but is not currently limited to, a feature that will allow users to communicate with each other through the use of a chat or video box, as well as the users specific tastes for films and T.V. to his or her profile that they can either make public or private. We believe that this will allow people to broaden tastes of genre and establish new relationships with other users based on their similar interests.

Lastly, the entire aesthetic design of the streaming service will be minimalistic and modern, which will provide the users with a streamline experience which will be both pleasing to the eye and simple to use.

Server Specifications

We will be using InMotion Hosting to host the website. Additionally we will be implementing Java, Javascript, and mySQL along with importing the Bootstrap library, of which contains HTML, CSS, and more Javascript. The website will be hosted on the domain name “videodivision.net”. All changes to the server will be tracked and logged by InMotion Hosting.

Functionality

1. Account Management

1.1 Functional Requirements

1.1.1 Signing up

Description: A user will be able to sign up using a valid email. They create a unique username and password for their account.

1.1.2 Signing in

Description: A user can sign in with a pre-existing account using their username and password.

1.1.3 Signing Out

Description: A user should be able to log out of their account at their own discretion.

1.1.4 Changing Password

Description: A user should be able to change their password by providing their old password and a new password

1.1.5 Changing email

Description: A user should be able to change their email by providing a new email and validating access to aforementioned email.

1.2 Non-Functional Requirements

1.2.1 Check for duplicate username

Description: When a user is creating their unique username, the system should check for duplicate usernames in the database to ensure uniqueness.

1.2.2 Check for valid email

Description: When a user is creating an account or changing their email, the system should send an email to the new address to verify the user's access

2. Payment/Billing

2.1 Functional Requirements

2.1.1 Billing

Description: Users will be charged for the basic package for 10 subscription slots. Users will also be charged for any additional fees if extra slots or movies are purchased.

2.1.2 Purchasing Slots

Description: Users should be able to purchase the basic package of 10 slots and pay an additional \$1.50 per extra slot. These subscriptions last for 30 days.

2.2 Non-Functional Requirements

2.2.1 Payment Details

Description: The website only accepts credit/debit cards as forms of payment

2.2.2 Payment Information

Description: A user must have a validated form of payment saved on their account for the service to auto-renew. Defaults on payments will lead to cancellation of the service.

3. Subscription Management

3.1 Functional Requirements

3.1.1 Subscribing to shows

Description: User should be able to subscribe to shows at their discretion for the 30 day period.

3.1.2 Purchasing Movies

Description: Users should be able to purchase a movie subscription at a rate of \$1 per movie.

3.1.3 Cancelling Subscriptions

Description: Users should be able to cancel subscriptions to shows during the last 5 days of the 30 day period.

3.1.4 Changing Subscriptions

Description: Users should be able to change subscriptions within the last 5 days of the 30 day period.

3.2 Non-Functional Requirements

3.2.1 Auto-renew Subscriptions

Description: The system should auto-renew a subscription to a show if the user chooses to take no action at the end of the 30 day period.

3.2.2 Track User Subscriptions

Description: The system should keep track of how many subscriptions the user has and the date that they expire.

4. Notification

4.1 Functional Requirements

4.1.2 Chat Notification

Description: The system will notify users of any chat invites and missed messages through email, user inbox or both on preference.

4.1.3 Subscription Notification

Description: The system will notify users of any new subscriptions through their email, user inbox, or both depending on preference.

4.2 Non-Functional Requirements

4.2.1 Email Duration

Description: Emails should be sent with a latency of no greater than 1 hour.

5. Media Storage/Indexing

5.1 Functional Requirements

5.1.1 Media Indexing

Description: The system will index all movies and shows with the use of metadata that is based on the content.

5.2.1 Searching

Description: The system will search through the index by using keywords that the user inputs into the search bar.

5.3.1 Storage

Description: The system will store selected movies and shows onto the database.

5.4.1 Removal

Description: The system will be able to remove stored videos onto the server database.

5.2 Nonfunctional Requirements

5.2.1 Metadata Type

Description: Metadata that is collected by the system will organize videos based on year, genre, actors, keywords, description, studio, etc

5.2.2 Search Pages

Description: Amount of movies and shows will be limited to 25 or 50 films or shows per pages.

6. Browsing Video List

6.1 Functional Requirements

6.1.1 Sorting videos

Description: The user can change the way the videos are sorted, such as by alphabetical order or by rating.

6.1.2 Searching videos

Description: The user can filter videos based on title, genre, subscription, cast, or director.

6.1.3 Changing tabs

Description: The user can switch between seeing the full video collection available on the service and their library of purchased videos.

6.1.4 Viewing video details

Description: The user can click on a video thumbnail to see information about the video, or play the video if the user holds a subscription to it.

7. Streaming Video

7.1 Functional Requirements

7.1.1 Selecting a video

Description: The user can select a video from their library of purchased videos, which will cause the video to begin playing.

7.1.2 Playing and pausing video

Description: The user can play or pause the currently selected media.

7.1.3 Seeking video

Description: The user can seek backwards or forwards in the video.

7.2 Nonfunctional Requirements

7.2.1 Automatic resume

Description: If the user leaves the page while a video is playing, the server should store the playback position and restore it when the user begins playing the video again.

8. User Profile Page

8.1 Functional Requirements

8.1.1 User Webpage

Description: A user will have a profile page that displays their profile picture, profile name, their watched shows, favorite shows, comments, and movie reviews.

8.1.2 Changing Account Picture

Description: A user will be able to click on their profile picture to upload a new photo as their profile picture.

8.1.3 Changing Profile Name

Description: A user will be able to change the name that displays on their profile. This is separate from their sign in information.

8.1.4 Adding Comment

Description: A user can add a comment that will show up on their profile page.

8.1.5 Editing Comment

Description: A user can edit a comment they made on their profile page.

8.1.6 Deleting Comment

Description: A user can delete a comment they made on their profile page.

8.1.7 Searching Users

Description: A user can search other users by their username or profile display name and access their profile page.

8.1.8 Adding Other Users as Friends

Description: A user can add another user as a friend via the other users profile page.

8.1.9 Removing Friends

Description: A user can remove friends from their account via their own profile page.

8.1.10 Opening Chat

Description: A user will be able to open a chat window to message a friend via their profile page. This chat will have all previous messages displayed in it.

8.1.11 Sending Message

Description: A user will be able to send a message to a friend via the chat window.

8.1.12 Favoriting A Show

Description: A user can favorite a show via the show's page.

8.2 Non-Functional Requirements

8.2.1 Updating Feed

Description: The newest movie ratings or comments a user makes will appear at the top of a user's profile page.

9. Rating Movies

9.1 Functional Requirements

9.1.1 Movie Review

Description: A user will be able to see movie reviews written by other users via a shows page. The review will have a rating system along with a text section. The average show rating will be displayed on the show page.

9.1.2 Create Movie Review

Description: A user will be able to write a movie review for a show and give it a rating.

9.1.3 Edit Movie Review

Description: A user will be able to change a review they have previously created via their profile page or show page.

9.1.4 Delete Movie Review

Description: A user will be able to delete a movie review via their user profile page or the show page.

9.2 Non-Functional Requirements

9.2.1 Populating Review Feed

Description: Reviews for a show will be updated to show the most recents first.

9.2.2 Viewing Friend's Review

Description: A user will be able to see their friend reviews separately from all the reviews for a show.

10. Security

10.1 Functional Requirements

10.1.1 Password Reset

Description: In the event of a user forgotten password, we may automate the password retrieval routine with either confirming security questions from time of registering or email verification.

10.2 Non-Functional Requirements

10.2.1 Invalid Login Strikes

Description: Track IP addresses attempting to login. For each invalid login, add an increment to the IP address, these will be considered strikes.

10.2.2 IP Blocking

Description: Coupling with Invalid Login Strikes, we can pass an IP address as *param 1* to deny login services to our website for *param 2* amount of *param 3* units of time measure.

10.2.3 Credit Processing

Description: Any credit processing in our services will be checked against the following logic:

Activations:

- How many credits does the user have
- Does the user have enough credits
- If user does not have enough credits, prompt for purchase.
- If use has enough credits, activate subscription.

10.2.4 IP Blocking

Description: Coupling with Invalid Login Strikes, we can pass an IP address as *param 1* to deny login services to our website for *param 2* amount of *param 3* units of time measure.

10.2.5 Invalid Email Entry

Description: Check entered email against current standard regex for valid email. Display error message if true, continue as normal if false.

10.2.6 Inbox Size Check

Description: Compute user's current inbox size and compare to administrator set maximum size. User will unable to receive messages if the next message received would result in a size greater than the maximum size of an inbox. Administrator notifications override this check.

11. Logging

11.1 Non-Functional Requirements

11.1.1 Number Of Visits

Description: Total views of both logged in users and non-logged in users to our services.

11.1.2 Number Of Unique Visitors

Description: Recorded number of hits per IP address basis

11.1.3 Number Of PageViews

Description: Recorded number of views for each piece of content provided by our services. We can collect favorable data.

11.1.4 Most viewed pages

Description: In contrast with *number of pageviews*, this recording will record which pages are the most visited ones sitewide.

11.1.5 Authenticated users and last time of authentication

Description: Amount of times a user has successfully logged in as well as their last time of successful login.

11.1.6 Web Browser Detection

Description: Increment into our database, based on unique IP addresses, which browser is most favored on our service.

11.1.1 Most Viewed Content

Description: Organizing the amount of viewed content by the following units of time:

- Today
- Week
- Month

- Year
- All Time

Maintenance Planning

In order to maintain the program once it is considered to be complete, we will be running automated scripts for various tests. The tests include but are not limited to:

1. Creating new users with randomly generated emails
2. Buying subscriptions, unsubscribing, and adding more than the initial 10 subscriptions
3. Watching multiple movies and shows at different time stamps
4. Testing the subscription renewal function of the website
5. Randomly personalizing the user homepage and using the chat function to interact with other users (not actual consumers)

and will continually check for the correct output of every function of the website.

The signees listed below agree to all the aforementioned requirements, allowing only the client Edwin Rodriguez, the ability to change/add/remove any given specification within reason. The final product must be delivered to the client by May 11th 2019.

Edwin Rodríguez, Client

Owen Dugmore, CEO of Infinity Bits