

Final Report

Aman Khoja
David Mathew
Minah Popal
Rohan Kozhikunnathu Samuel
Justin Yim

Group 12

Executive Summary:

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

Executive Summary:	2
List of Figures:	5
List of Tables:	5
Introduction	6
Purpose and Scope	6
Product Overview (including capabilities, scenarios for using the product, etc.)	6
Structure of the Document	6
Terms, Acronyms, and Abbreviations	6
Project Management Plan	7
Project Organization	7
Life Cycle Model Used	7
Risk Analysis	8
Hardware and Software Resource Requirements	8
Deliverables and Schedule	9
Monitoring, Reporting, and Controlling Mechanisms	9
Professional Standards	9
Evidence all the artifacts have been placed under configuration management	10
Impact of the project on individuals and organizations	11
Requirement Specifications	11
Stakeholders for the system	11
Use case model	11
Rationale for your use case model	14
Non-functional requirements	15
Architecture	17
Architectural style(s) used	17
Architectural model	19
Technology, software, and hardware used	20
Rationale for your architectural style and mode	22
Design	23
GUI (Graphical User Interface) design	23
Static model	25
Dynamic model	27
Rationale for your detailed design model	27

Traceability from requirements to detailed design model	28
TestPlan	30
Requirements/specifications-based system level test cases	30
Test Case 1 (Table 2)	30
Test Case 2 (Table 3)	31
Test Case 3 (Table 4)	32
Test Case 4 (Table 5)	32
Test Case 5 (Table 6)	33
Test Case 6 (Table 7)	34
Test Case 7 (Table 8)	34
Test Case 8 (Table 9)	35
Test Case 9 (Table 10)	36
Test Case 10 (Table 11)	36
Traceability of test cases to use cases	37
(Table 12)	37
Techniques used for test generation	38
Assessment of the goodness of your test suite	
(Which metrics were used for such assessment?)	40
Acknowledgment References	40

List of Figures:

1. Figure 1: Life Cycle Model Used - Agile
2. Figure 2: Project Management Plan Configuration Management
3. UC_1: Use case 1 for one of the functional requirements.
4. UC_2: Use case 2 for another functional requirement.
5. UC_3: Use case 3 for another functional requirement
6. SC_1: Whole view of the system.
7. 4.1a - Client Server Architecture
8. 4.1b - Business context diagram of our system where some of the scope that deals with our system is defined.
9. 4.1c - System context diagram of our system that explains the different entities that will be interacting with the system.
10. 4.2a - Video encoding and decoding process used in our system.
11. Business Case 1: User will interact with the the About and Registration Page
12. Business Case 2: User will be able to go from the homepage to the Multi Screens
13. Business Case 3: Once the user is on the Selection screen and a flash occurs on the smaller screen then that screen will be set to the main big screen
14. 5.1a Front Page
15. 5.1b Selection Screen
16. 5.1c Watching Screen
17. 5.2a Class Diagram
18. 5.2b Backend Server
19. 5.3a Sequence Diagram
20. 5.4a Text Editor - Backend

List of Tables:

1. Project Organization
2. Test Case 1
3. Test Case 2
4. Test Case 3
5. Test Case 4
6. Test Case 5
7. Test Case 6
8. Test Case 7
9. Test Case 8
10. Test Case 9
11. Test Case 10
12. Traceability Matrix Table

1. Introduction

SE 4485 Software Engineering Project Final Report

1.1. Purpose and Scope

To create a video streaming prototype (REDflix) that allows users to watch and experience multiple live events happening on one single screen. Avoid the hassle of switching channels, this service will provide the tools to watch multiple entertainment events on the go. Purpose of the project is to influence the thinking of our product management and architecture teams regarding current state-of-the-art technologies.

1.2. Product Overview (including capabilities, scenarios for using the product, etc.)

This product has the capabilities for a user to login or register to login. It provides users to have multiple video streaming services in one platform. There are different packages that a user can purchase based on their interests. While the user is browsing content to watch, they can select content from any service they subscribed to and are able to watch up to 6 videos.

Scenarios for using this product would be paying to an individual service every month instead of multiple accounts across different platforms. It's convenient to have all the videos you want to watch in one place. Say if a user is interested in watching a live streaming of a football game, but nothing was happening they could also pull up a tv show and watch that as the game is happening.

1.3. Structure of the Document

This document contains a compilation of all the deliverables from the project management plan, requirements document, architecture document, design document, and lastly the test plan.

1.4. Terms, Acronyms, and Abbreviations

- UC - Use Case
- HTML - Hypertext Markup Language
- CSS - Cascading style sheet

2. Project Management Plan

2.1. Project Organization

Phase	Members	Description
Requirements	Aman, Minah, Justin, David, Rohan	Create a requirements document for the video streaming prototype
Designer	Aman, David, Minah	Design UI Screens which users with interact with
Back End	Aman, David, Minah	Design the back end in Amazon AWS, create procedures and queries for the home screen.
Front End	Justin, Aman, David, Minah, Rohan	Create a web host service using HTML and design and implement all the UI Screens for the project
Testing	Aman, Roha, Justin	Testing the UI screens and video service prototype
Deployment	Aman, David, Minah, Justin, Rohan	Release the final version of the project

Table 1

2.2. Life Cycle Model Used

We will use the agile, to make room for any sudden changes throughout the project lifecycle. We choose agile because even though this is not a super complex project we still want to make room for any type of incoming features we might want to add.



Figure 1

2.3. Risk Analysis

2.3.1. Communication Risk

2.3.1.1. Probability: Low

2.3.1.2. Impact: Medium

2.3.1.3. Reason: All the team members live close to campus. In case of an emergency, it is possible for the group members to meet face to face. In normal circumstances, the team will be communicating remotely.

2.3.2. Process Risk

2.3.2.1. Probability: Low-Medium

2.3.2.2. Impact: Medium-High

2.3.2.3. Reason: Errors identified later in the design process have a large impact. Process issues increase time risks.

2.3.3. Time Risk

2.3.3.1. Probability: Medium

2.3.3.2. Impact: High

2.3.3.3. Reason: This is a complex problem to solve in sixteen weeks. It is absolutely essential to manage the limited time available for the project. Being behind schedule is potentially fatal to the project.

2.3.4. Complexity Risk

2.3.4.1. Probability: Medium

2.3.4.2. Impact: Medium-High

2.3.4.3. Reason: The project components are relatively complex and require team members to spend time researching, and learning techniques and technologies new to them before implementing the system.

2.4. Hardware and Software Resource Requirements

2.4.1. Hardware:

For our group our hardware resources will be using 1 Windows 10 laptops which together will have a minimum of 8 GB of RAM so running our programs will be relatively fast and effective. The other 4 members have apple laptops that are both just as fast as the Windows 10 laptops. For the sake of consistency we all will do our code separately on our own laptops and when we need to combine certain things then we will use the ECS computer lab and use their Windows 10 computers.

2.4.2. Software:

The compilers and types of software that we will be different depending on what phase of the project that we are on. Because this will be a A streaming video service we will most likely use HTML and CSS for the structure and as well as Javascript to make the website user friendly, that will be on the front end. For the back end we will probably use SQL so the video streaming will be smooth. All of these software requirements are provided in the ECS computer lab.

- Aman: Django Framework 2.0, Python 3.7, SQLite
- David: Codepen, git commands, MYSQL, JavaScript/CSS/HTML
- Minah: Codepen compiler, Django framework, git commands, SQL operations, JavaScript/CSS/HTML
- Rohan: Django, Codepen, SQLite, git commands, JavaScript/CSS/HTML
- Justin: Django Framework 2.0, SQLite, git commands, Python 3.7

2.5. Deliverables and Schedule

Project Management Plan	Due 01/31, Friday
Requirements Documentation	Due 02/14, Friday
Architecture Documentation	Due 02/28, Friday
Detailed Design Documentation	Due 03/13, Friday
Testing Plan	Due 04/10, Friday
Final Project Report	Due 04/30, Thursday
Final Project Demonstration	Due 04/30, Thursday

2.6. Monitoring, Reporting, and Controlling Mechanisms

2.6.1. Weekly Meeting

The team shall conduct weekly meetings in order to discuss and update each other on the progress of the project.

2.6.2. Documentation and Code Repositories

The team will be using Google Docs and Github in the primary stages of the project and if needed, will use different software/services in the later stages of the project.

2.6.3. Team Lead

The team lead will be monitoring team members' work and will be responsible to submit different parts of the projects on time.

2.7. Professional Standards

2.7.1. Integrity

All research is expected to have academic integrity, references shall be made with IEEE format.

2.7.2. Research

Each team member is expected to contribute at least two hours weekly of their time for the purpose of this project. This is documented on our weekly status excel sheet.

2.7.3. Due Dates

Each team member's tasks must be completed on time for each deliverable.

2.7.4. Team Meetings

Team members are supposed to show up face to face or remotely when a team meeting is scheduled.

2.8. Evidence all the artifacts have been placed under configuration management

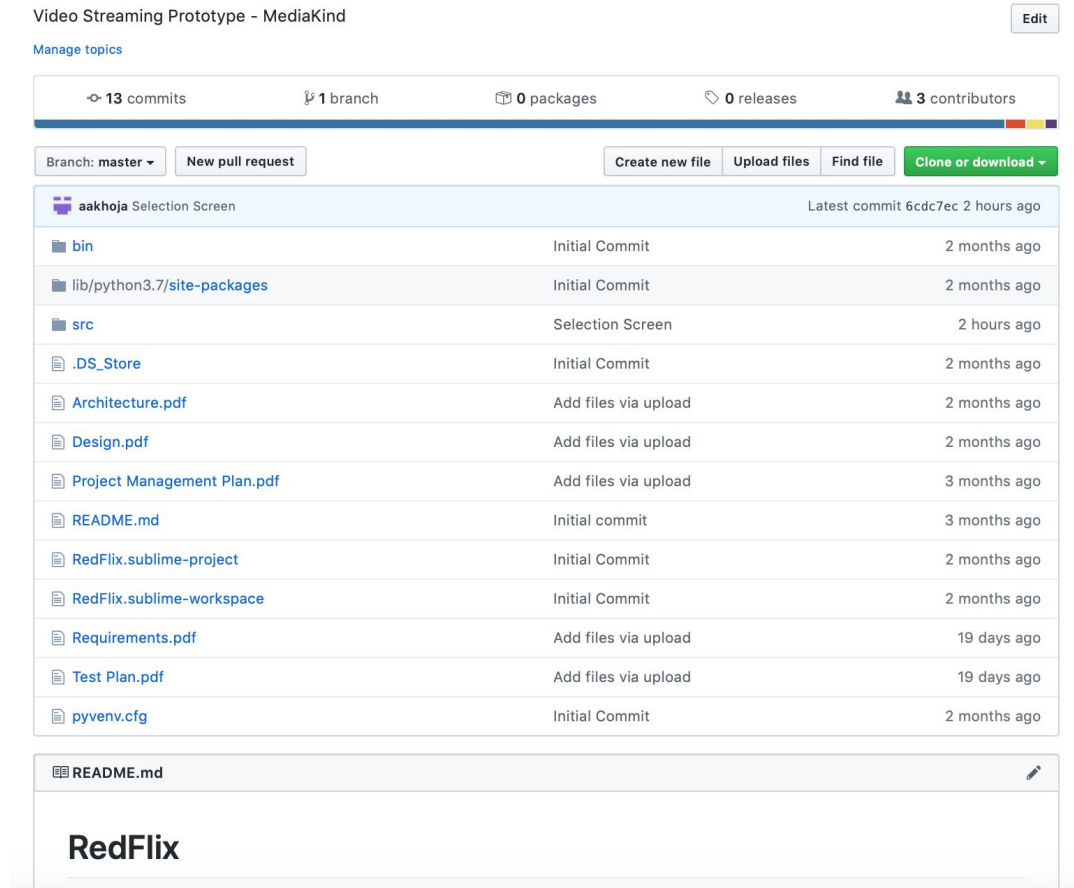


Figure 2

2.9. Impact of the project on individuals and organizations

The ability to watch multiple video media content from various streaming services, featuring the ability to view up to 6 videos simultaneously on the same screen.

3. Requirement Specifications

3.1. Stakeholders for the system

MediaKind
Professor W. Eric Wong
TA Linghuan Hu

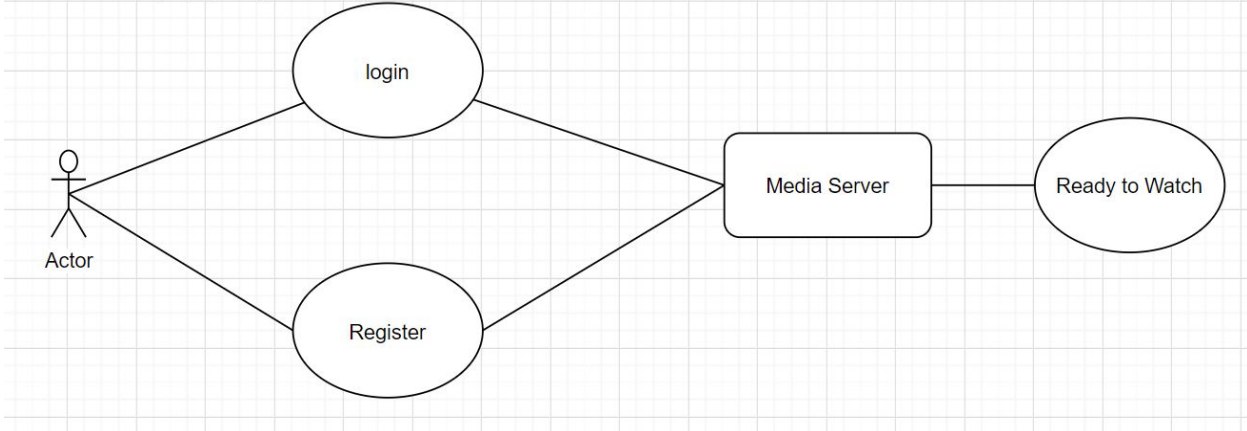
3.2. Use case model

3.2.1. Use case name: UC_1- Login & Registration

Participating Actors: User, Media Server

Scenario: New users registering or if already registered, logging into the system.

User Functions (US_1): Login and Registration

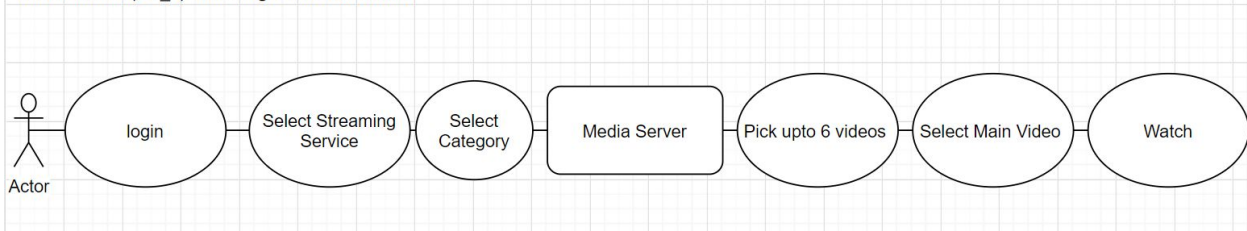


3.2.2. Use Case name: UC_2 - Selecting content to watch

Participating Actors: User and Media Server

Scenario: Once the user logs in, the user may browse our service. In this use case, it shows how the user can start watching videos. First the user logs in, then proceeds to select a streaming service and type of video to watch from. Then the search result is sent to the media server to provide output to the user. After this the user can select up to 6 videos to project and can choose one main video out of 6 to emphasize it.

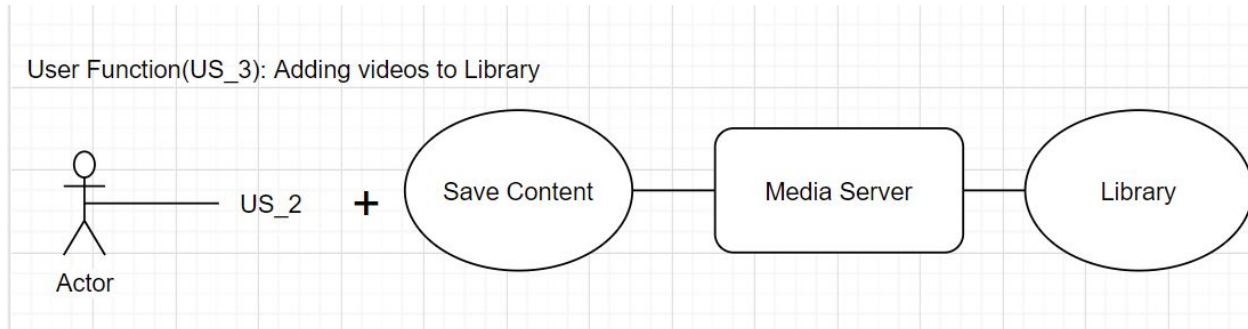
User Function (US_2): Selecting Content to Watch



3.2.3. Use Case name :UC_3- Adding Videos to Library

Actors: User, Media Server

Scenario: This case shows how the user can save a video to the library. This step involves all the steps in UC_2 and then from the watch screen, the user can add the videos by an option provided. These videos will be sent to the media server to save for later and the user can access it via the library option.



3.2.4. Use Case name :UC_4 - Video Functionality:

Actors: User

Scenario: User will select a video to watch. Once video is selected they shall be able to fast forward 10 seconds, rewind 10 seconds, pause, adjust volume, adjust video quality, and change subtitle and language to the available options according to their preferences.

3.2.5. Use Case name :UC_5 - Transitioning from selection screen to main:

Actors: User

Scenario: The user transitions from the selection screen to the main screen. The user will select one of the videos from the list 6 videos and after clicking one of the videos, the user will be taken to the next screen and that screen will display all the 6 videos and the video the user selected will be emphasized.

3.2.6. Use Case name :UC_6 - User authentication and authorization:

Actors: User, Media server, Database

Scenario: In this scenario, the user authentication is done by using azure backend cloud servers to authenticate users with correct login and subscription plans.

3.2.7. Use Case name :UC_7- Media platform authentication:

Actors: User, Media server, Third party content provider

Scenario: The user will be authenticated to use the media platforms as they have selected their plans.

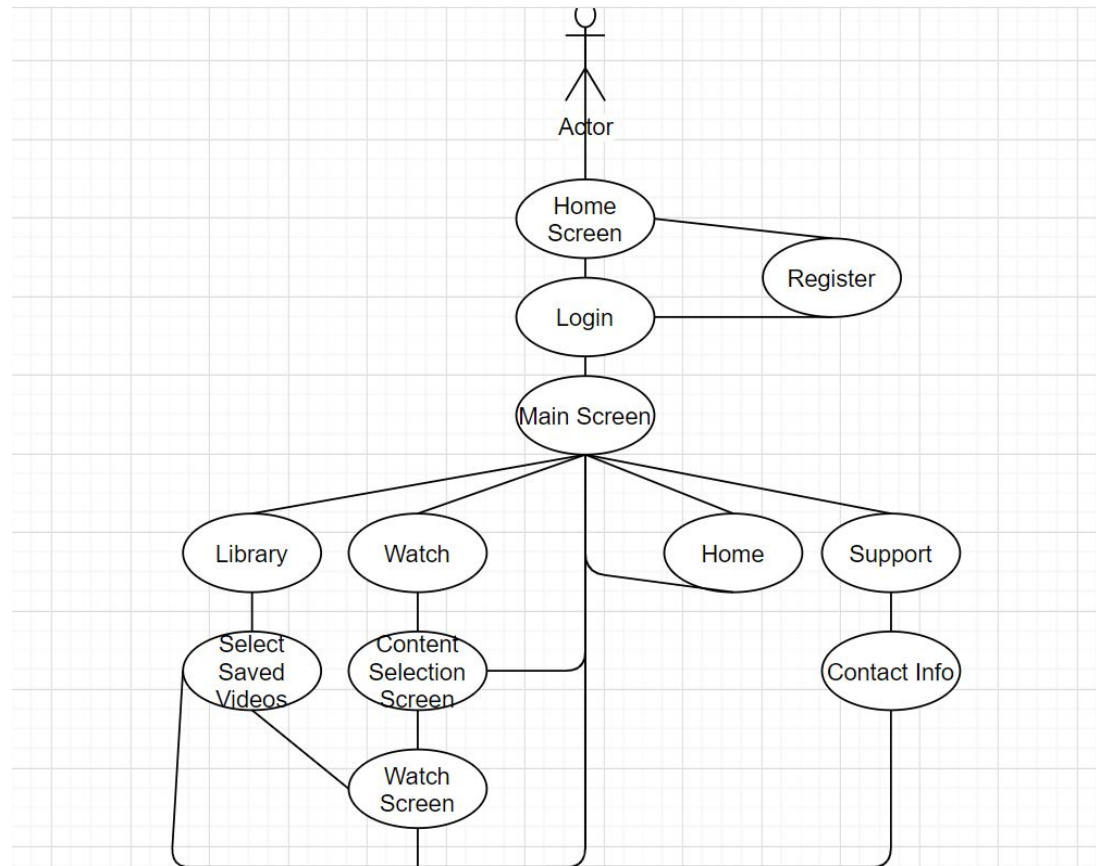
3.2.8. Use Case name :UC_8 - First time user enters Homepage:

Actors: User

Scenario: Upon entering the website, without the need of a user to be logged on or registered, the user shall be able to browse around. 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. This will give them a feel for the website and give them a chance to see what Redflix has to offer. Result of this will allow the user to login or want to register for a subscription to Redflix.

3.2.9. System View: SC_1

Scenario: The user will initially be shown the home screen listing what services we offer. The home screen will provide the user to login or if new, the option to register for service. Once the user logs in, the user can select from the library, watch, home or support option. The library option will let the user watch the saved content he/she has saved before. The watch option will take the user to another screen where the user will select the content to watch. In this screen the user can select from which platform he/she wants the videos streamed from and also select the category of video. The user can select upto 6 videos to play simultaneously on one screen and then select one of those videos to make stand out from others. Once the user selects which video to stand out, the final watch screen will be displayed with all 6 videos. All of these sectors will have a home button to get back to the homepage.



3.3. Rationale for your use case model

- 3.3.1. A user shall be able to log in
- 3.3.2. A user shall be able to click a Support tab despite membership status
- 3.3.3. A user shall be able to click an About Us tab despite membership status
- 3.3.4. A user shall be able to register via facebook, google, or preferred email
- 3.3.5. A user shall be able to select a subscription plan
- 3.3.6. A user shall be able to browse available content
- 3.3.7. A user shall be able to search available content
- 3.3.8. A user shall be able to select videos for playback on six screens simultaneously
- 3.3.9. A user shall be able to expand one of six selected videos
- 3.3.10. A user shall be able to switch expansion of either six screens
- 3.3.11. A user shall be able to organize selected videos/categories
- 3.3.12. A user shall be able to stream videos
- 3.3.13. A user shall be able to fast-forward videos
- 3.3.14. A user shall be able to pause videos
- 3.3.15. A user shall be able to rewind videos
- 3.3.16. A user shall be able to scrub videos
- 3.3.17. A user shall be able to adjust stream volume
- 3.3.18. A user shall be able to access the home screen at any point

- 3.3.19. A user shall be able to log out
- 3.3.20. A user shall be able to unsubscribe
- 3.3.21. A user shall be able to edit payment information
- 3.3.22. A user shall be able to add videos to a Watch list
- 3.3.23. A user shall be able to remove videos from watch list
- 3.3.24. A user shall be able to stream videos from supported platforms
- 3.3.25. The system shall be able to accept payments via credit/debit card
- 3.3.26. The system shall show a description of the platform when the mouse hovered over that specific platform
- 3.3.27. The system shall redirect the user to the Watch screen when the user clicks on a platform in the Main screen
- 3.3.28. A user shall be able to contact customer service in the Support tab via email/phone number
- 3.3.29. The system shall display FAQs in the Support tab
- 3.3.30. A user shall discover company details in the About Us tab
- 3.3.31. The system shall offer a search service based on platform, category, and or text search
- 3.3.32. The system shall stream search results to screens based on search order
- 3.3.33. The system shall exit the full screen mode via the esc key
- 3.3.34. The user shall be able to rearrange screens via drag and drop
- 3.3.35. The user shall be able to collapse and expand the dashboard

3.4. Non-functional requirements

- 3.4.1. Security
 - 3.4.1.1. The system shall be secure with 99.9% certainty
 - 3.4.1.1.1. The system shall accomplish this by outsourcing all security related implementations and problems to another software company
 - 3.4.1.2. The user data will be stored securely on a cloud based system.
 - 3.4.1.2.1. The cloud based storage should provide about 200TB worth of storage.
 - 3.4.1.3. The system shall require users to use a password that is 8-20 characters long
 - 3.4.1.3.1. The system shall require at least 1 uppercase letter, 1 lowercase letter, and 1 number
 - 3.4.1.4. Access rights authorizing on who is allowed to watch both live as well as on demand streams has to be possible on per stream basis (AD group based or individual AD members)
 - 3.4.1.4.1. The system shall only authenticate users if the subscription is paid in full and is not breaking the policies set by Redflix standards.

3.4.2. Availability

3.4.2.1. The system shall be available 24/7

3.4.2.1.1. If a user forgets their login credentials the system will send a reset password email within 4 seconds to the accounts email address

3.4.2.1.2. The streaming system must be designed and delivered in high availability configuration with built in redundancy to ensure smooth operation even when one component fails.

3.4.3. Scalability

3.4.3.1. The system shall be able to reach over 100,000 users

3.4.3.1.1. The media server should be able to withstand a minimum of 100,000 concurrent users at a given time.

3.4.3.2. The Audio Video Streaming System has to be scalable for up to 2000 simultaneous users from within the IAEA at all locations plus up to 500 additional but external users connecting through the Internet.

3.4.3.2.1. For connection of up to 500 external users, Cloud based services are allowed but all internal connections have to be streamed from on premise systems.

3.4.3.3. The systems bandwidth shall have a minimum bandwidth of 65 Mbps

3.4.3.3.1. This shall be best achieved if the user has a fiber optics connection or ethernet cable.

3.4.4. Reliability

3.4.4.1. When the user registers and sets up an account with the system their private information secure with us

3.4.4.1.1. Their information will be available to them upon the request of the user request

3.4.4.1.2. If the user desires to cancel their subscription with the system then the system will delete their credit card information but keep their name, email, and phone number.

3.4.4.1.2.1. After 2 years of inactive use from the user, the system will permanently delete all records from the system

3.4.4.2. The user must be able to log into the system with their credentials (email and password)

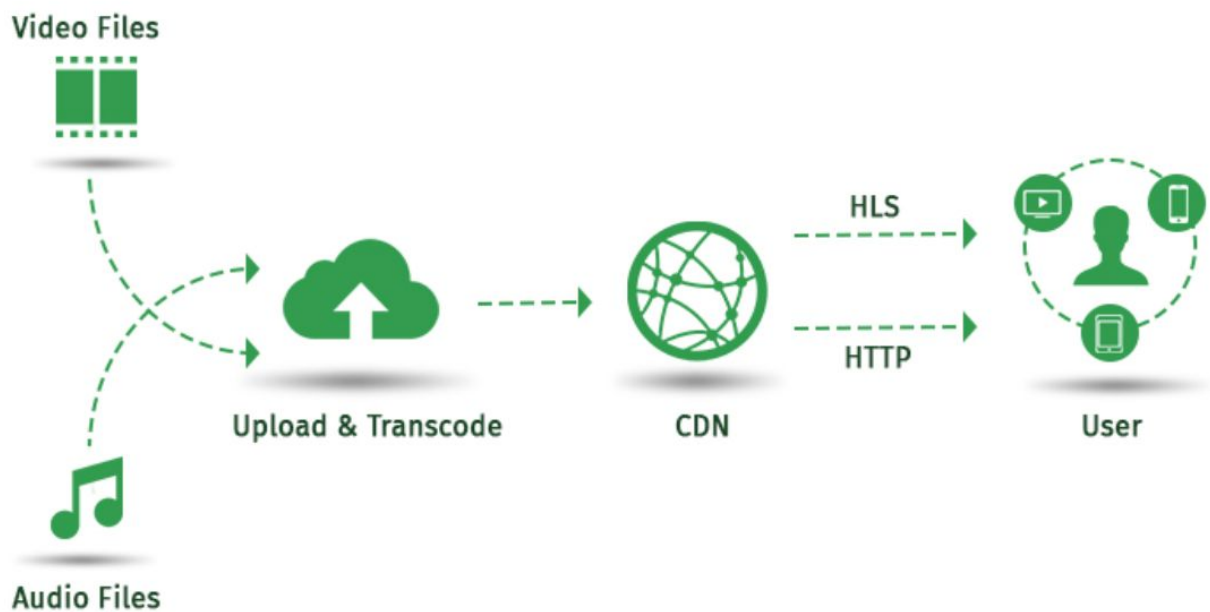
3.4.5. Usability

3.4.5.1. Front end

- 3.4.5.1.1. The system shall clearly display a button/click option for the home page
- 3.4.5.1.2. The system shall showcase a drag-and-drop feature when ordering currently playing media
- 3.4.5.1.3. The system shall clearly display a search feature
- 3.4.5.2. Back end
 - 3.4.5.2.1. The system will use AWS to store customer data in a secure manner that is easily accessible to the user upon request
 - 3.4.5.2.2. Languages that the system will use for backend will be Javascript

4. Architecture

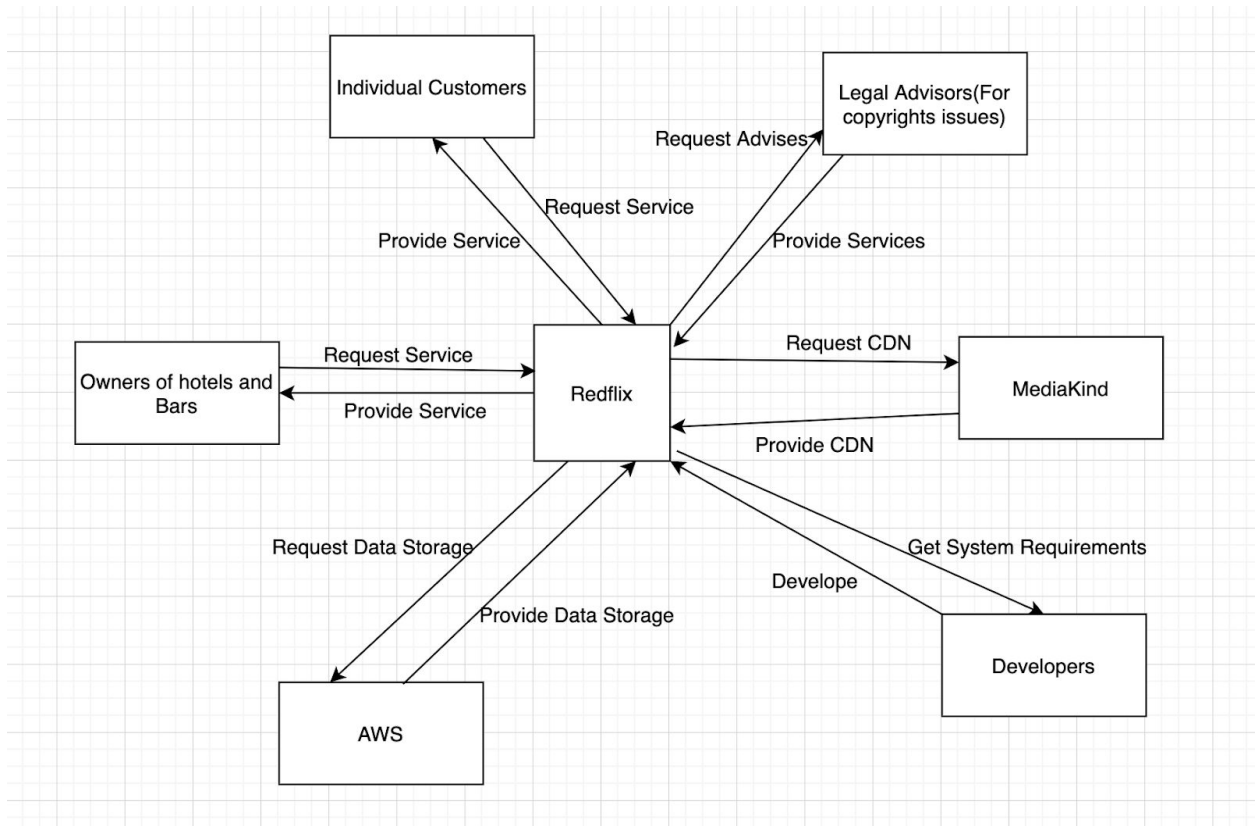
4.1. Architectural style(s) used



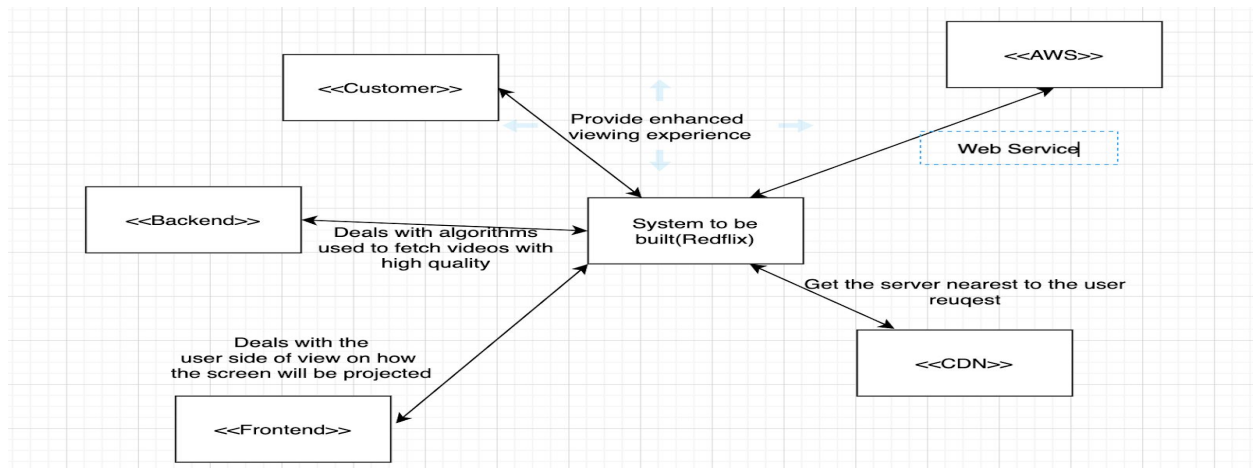
4.1a - Client Server

For this project, we will be using the client server architecture. The clients will be accessing the server every time they use our service and the server will answer the user's request. The server will be fetching videos and audio files

4.1b Business Context:



4.1c System Context:



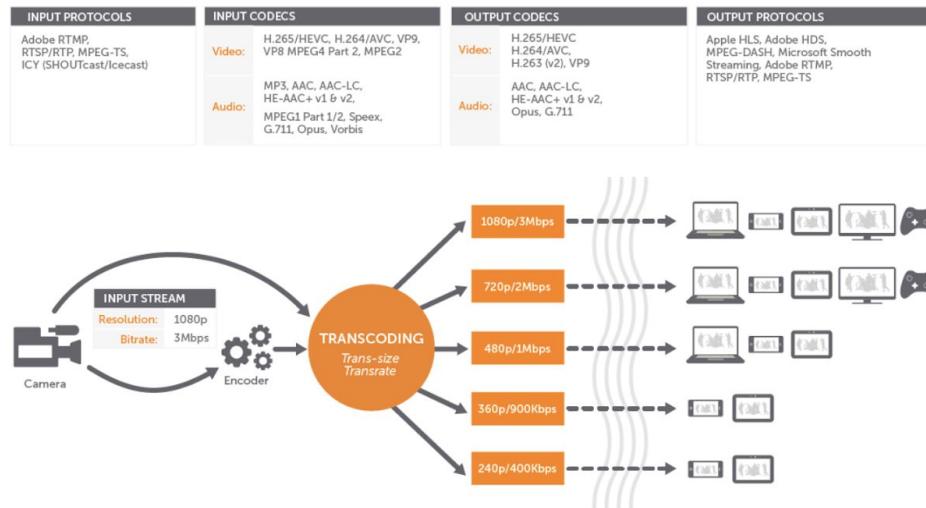
4.2. Architectural model

Redflix will be buying distribution rights from major content delivery organizations like Netflix, Hulu, Disney+, Prime Video, Live Channels, etc. We will support many more in the future as we grow. Contents given by these organizations will then be stored to our databases around the world. With contents from over 60+ distributors you won't miss out on any of your favorite entertainment. We will be providing our valuable service with various different packages from all over the globe.

Our contents will be stored on Amazon S3 database web services. The database will store our key information regarding our users, security protocols and authenticating and verifying the encoding done by Amazon on the EC3 server. The transcoding will be done on Amazon EC2 servers. Amazon provides 99% reliability so you can watch all your favorite entertainment without any fear of content not being available. Once a user has logged in, a server request is sent to our CDN network. CDN network that connects to our main server which stores all the contents for Redflix. The CDN then responds by connecting the user with the closest data server by their geo location and CDN then produces the content from the closest network server.

The amount of content that Redflix is offering we have partnered up with multiple different CDN providers like Akamai CDN, Amazon Cloudfront and Google Cloud CDN. These CDN's will provide content to users worldwide in an instant. With our own Redflix Open Connect technology, CDN networks can connect to our open connect system and access all the different types of content that we provide. The data is accessed once authentication is validated and then the content is delivered to the user. Our CDN's are protected and secured at all times. The security will be provided by our CDN partners on our contents and for all the database storage required in the Amazon EC2, EC3 and web services.

Redflix will deliver high quality content depending on your internet provider and speed. Diagram 3a. Describes how Redflix optimizes the videos to produce the best quality content for its user. Currently, we only operate on the web but future iterations will allow the users to stream and watch all these favorite entertainment in different portable devices like cell phones, tablets. The idea of having multiple UI screens which enables users to watch multiple different content under one screen.



4.2a Video encoding and decoding process

4.3. Technology, software, and hardware used

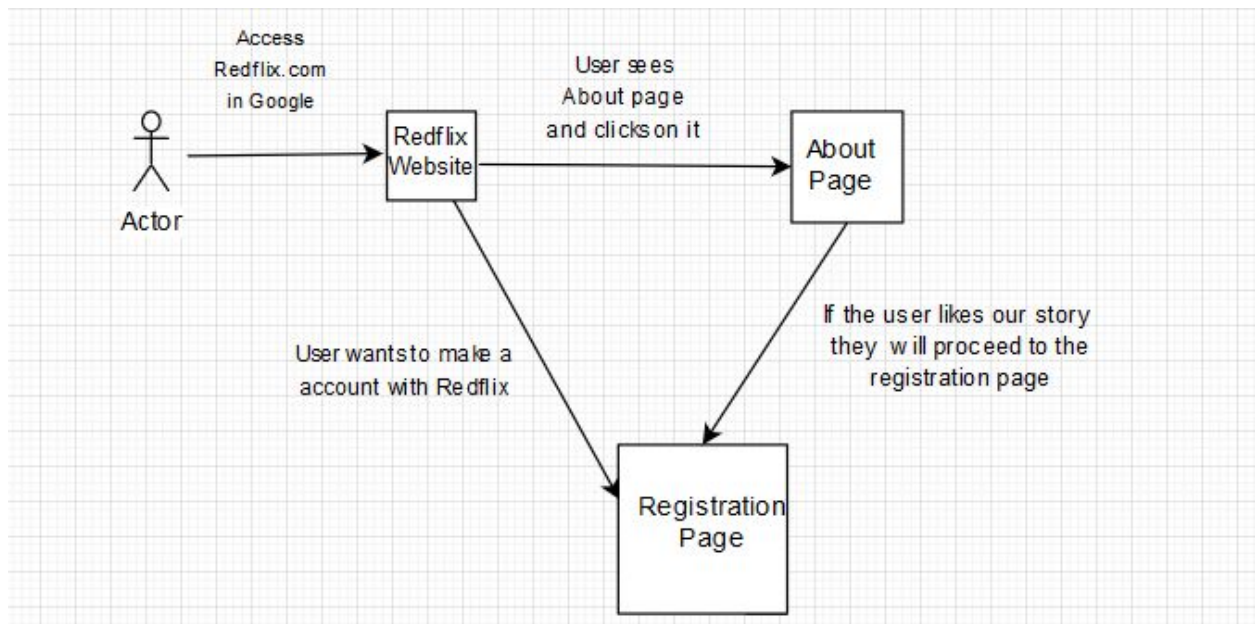
Software:

Java, HTML, JavaScript, Codepen, DB Browser for SQLite

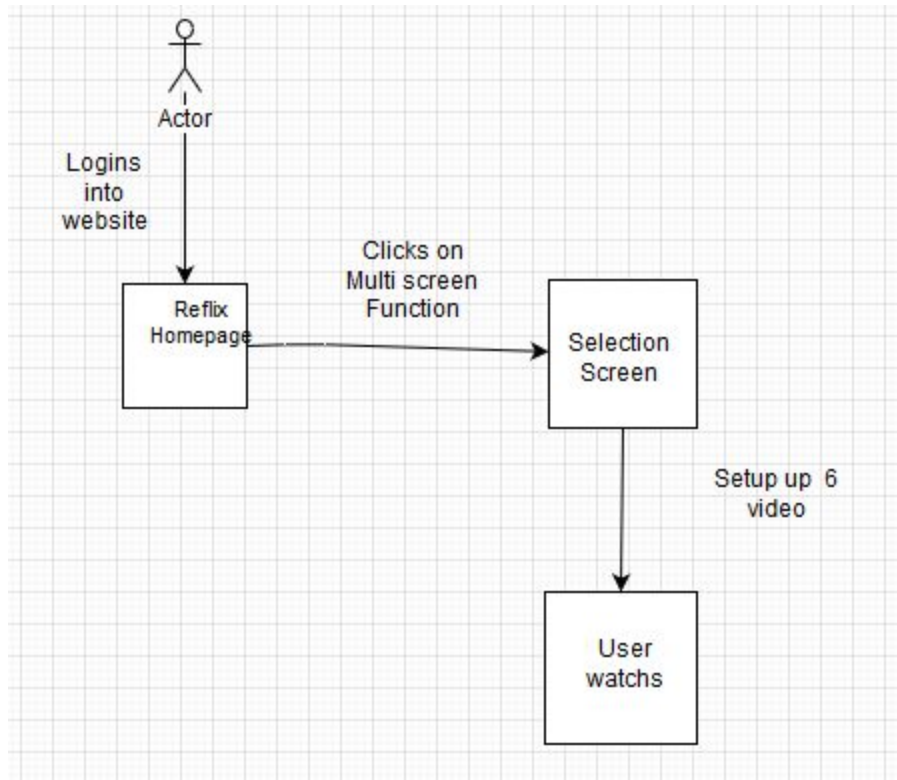
Hardware:

Laptops

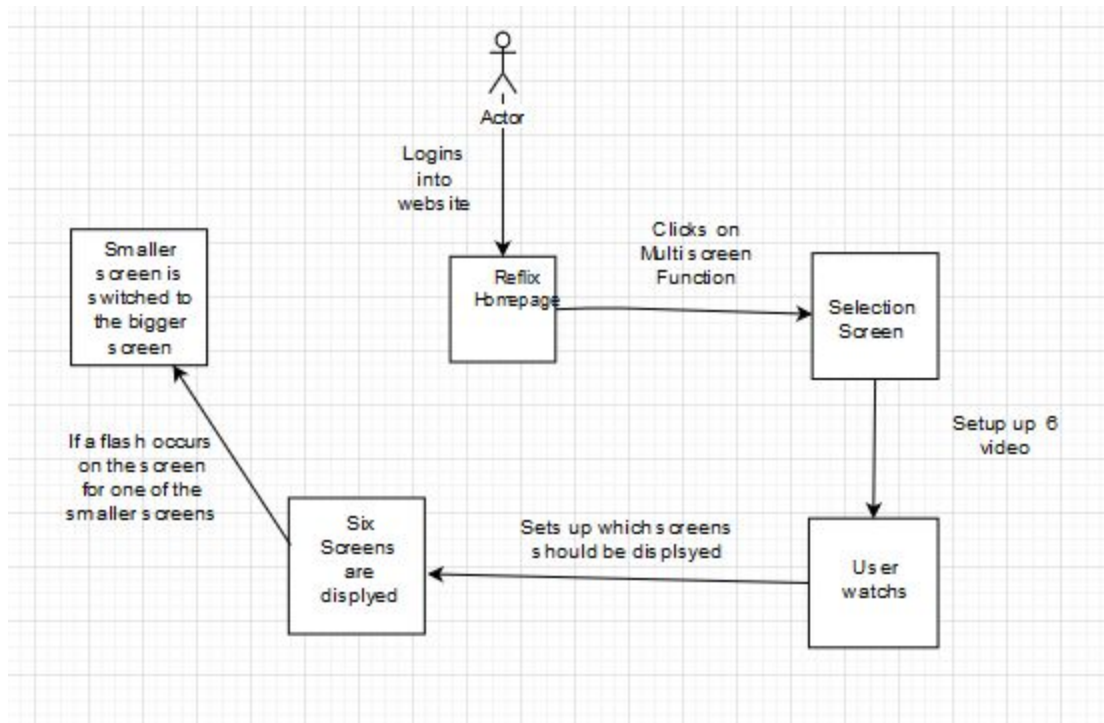
Business Case 1: User will interact with the the About and Registration Page



Business Case 2: User will be able to go from the homepage to the Multi Screens



Business Case 3: Once the user is on the Selection screen and a flash occurs on the smaller screen then that screen will be set to the main big screen



4.4. Rationale for your architectural style and mode

4.4.1. 6 screens all being streamed at the same time :

Today Bars and Restaurants are big places for people to watch if there is a big event happening. Now, there could be multiple events happening at the same time. Currently, the TV's in these places can only show one event at a time. If there are multiple groups of people who might prefer to watch different live events. This is where our software comes in. Our software will not only be limited to restaurants and bars but also to an individual level. Anyone can subscribe to our service and can utilize watching multiple events at the same time.

4.4.2. Video motion detection when identifying something "important" :

While the user is watching all six screens or if the user is looking away from the computer screen, video motion detection will project the video with the most action. This will attract the users attention, for example, if a live streaming of a football game was happening and a team was about to make a touchdown, that would be projected so the user has its attention on that current screen

4.4.3. Switching screens to different content of your choice :

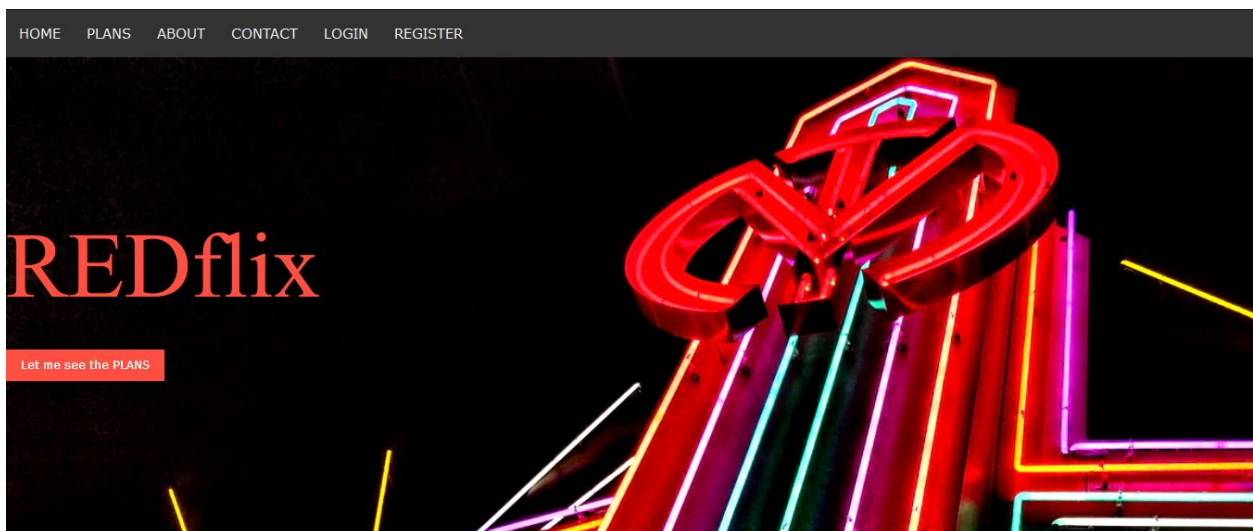
People tend to have short attention spans, therefore being able to change the content of your screens constantly to something new is satisfying for a user. If a user does not like the broadcaster for a certain sports game, he or she can change it to another channel for another spokesman.

5. Design

5.1. GUI (Graphical User Interface) design

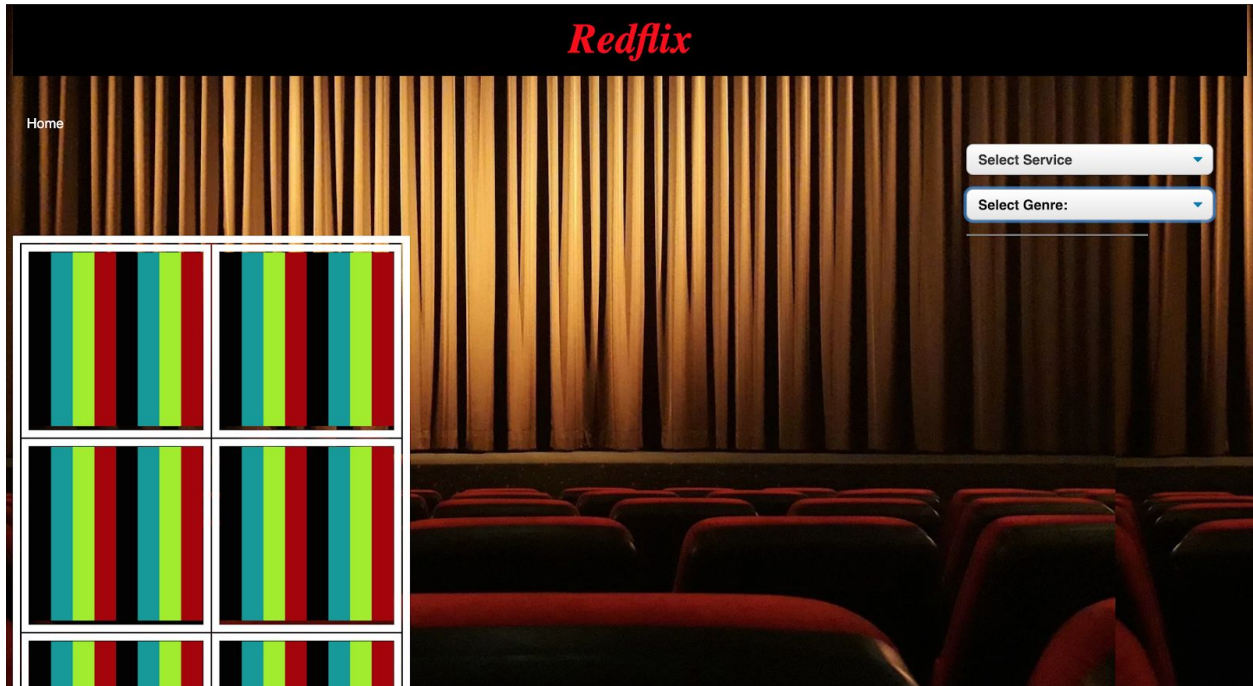
Front Page

This is a screenshot of the front page of our website, here the user will have access to all the menu options in the navigation bar as well as the actual platform itself.



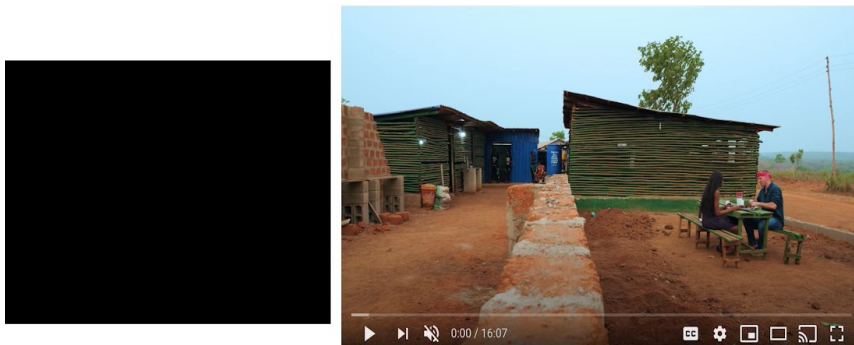
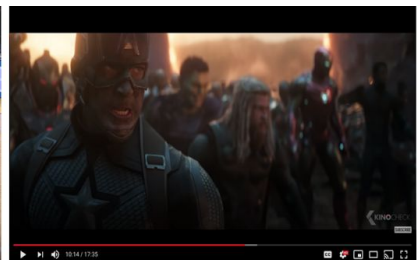
Selection Screen

The selection screen where the user can select the type of content provider and the video categories.

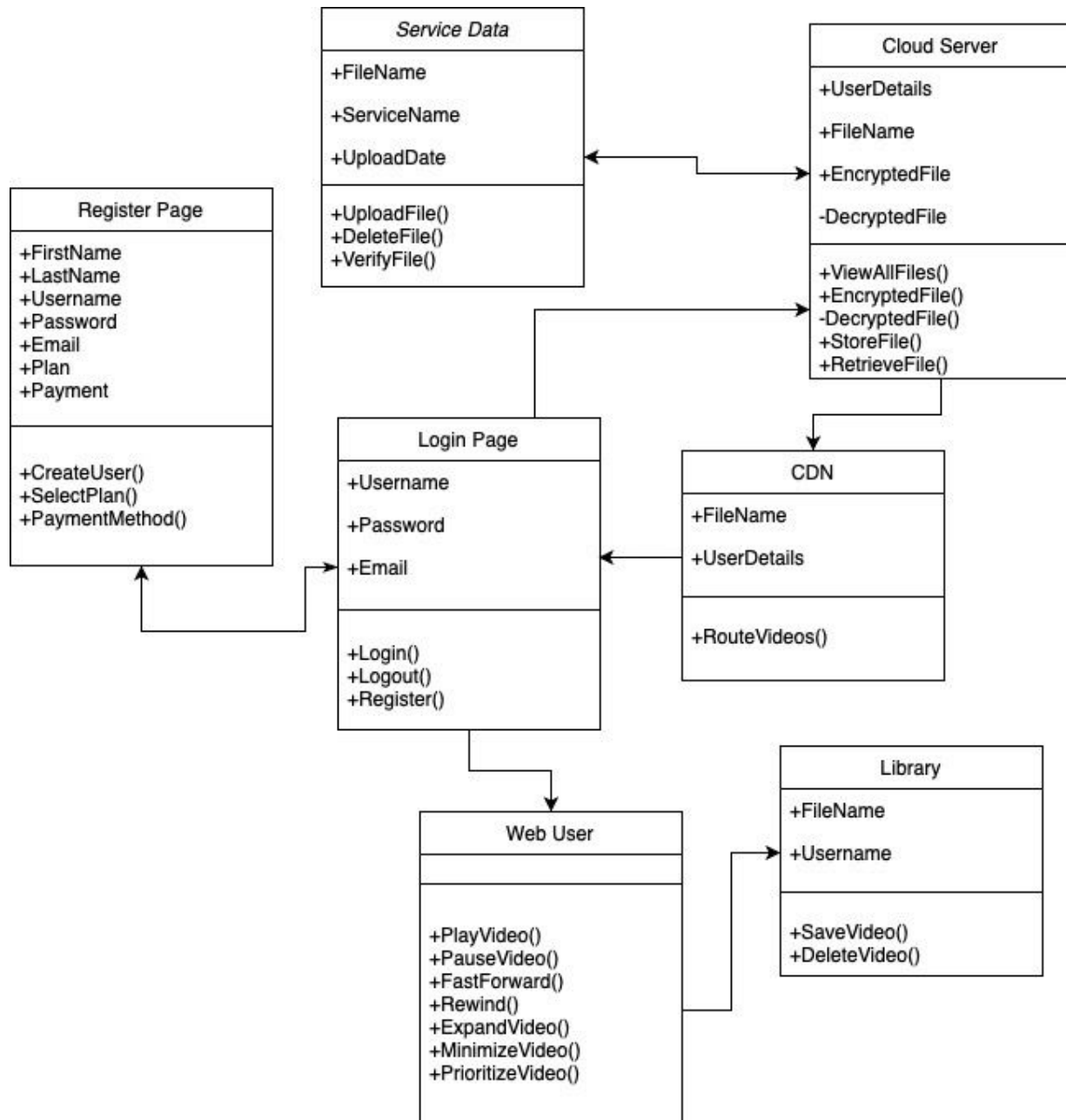


Watching Screen

The main viewing screen, where all six videos are projected and the main video screen which is the bigger one.



5.2. Static model



5.2a Class Diagram

Service Data: Assuming the content from the services (ex: movies, youtube, live streams, etc) have those methods.

Register Page: This page is where the user will be able to choose a payment plan for the type of service that they want. As well as enter their email, name, password, and any

other essential data that will be needed for the registration process. When their credit card is accepted then they will go back to the login page

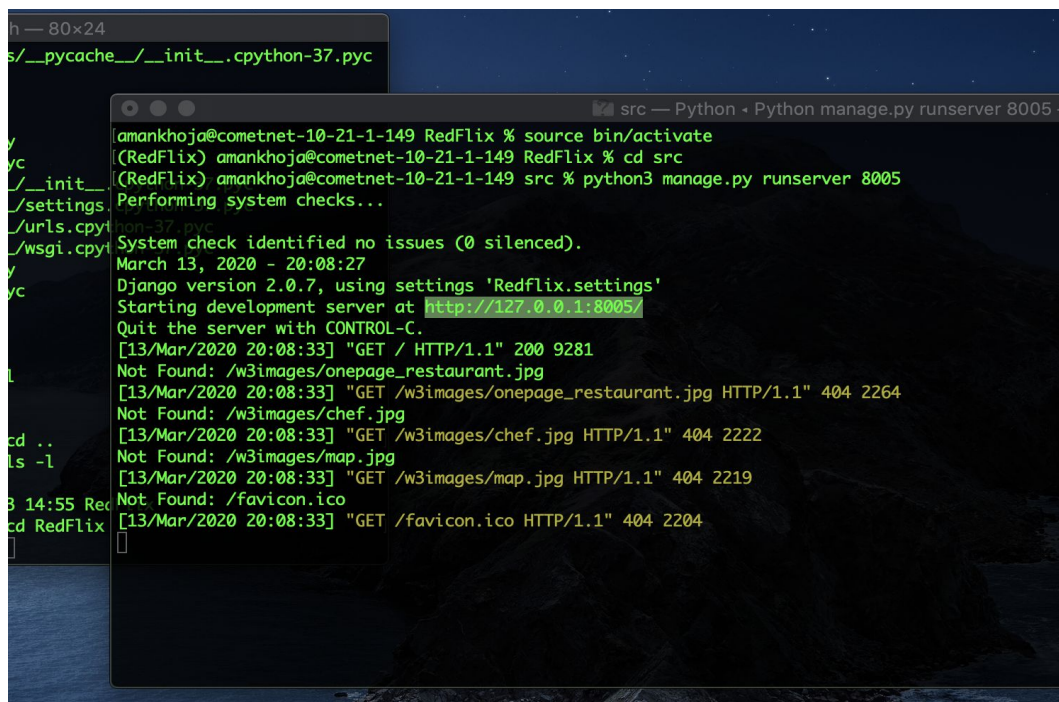
Login Page: If the user does not have an account with us then they can click on the register button to go to the register page. Here the user just needs to input the username and email to get access to the website.

CDN: Content Delivery Network, a system for delivering pages along a network. This will deliver content based on the user's geographical location.

Web User: Authorized user now has access to the different functionalities of video play

Library: User can save their favorite videos and or delete them from their library

Cloud Server: Not a physical server but a virtual one and built and hosted on the cloud. Contains things like FileName, UserDetails, and RouteVideo.



```
h — 80x24
s/_pycache_/__init__.cpython-37.pyc

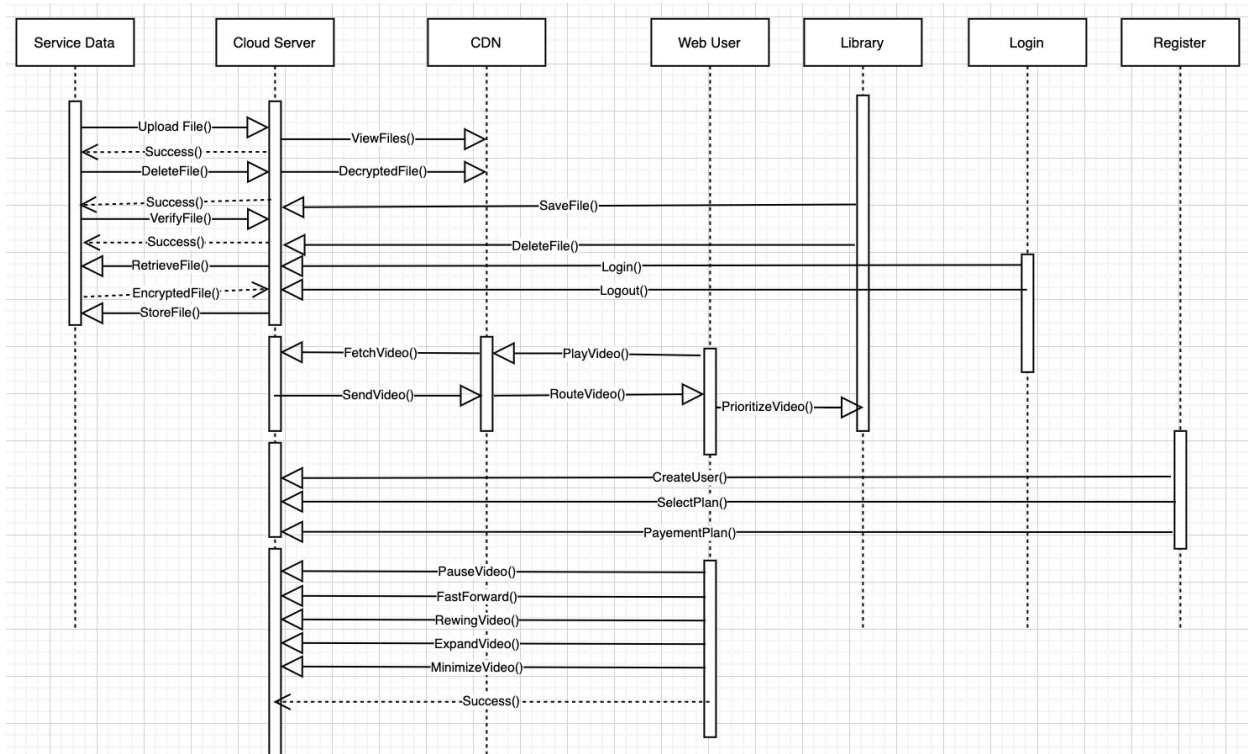
y
yc
/_init_
/_settings
/_urls.cpy
/_wsgi.cpy
y
yc

l
cd ..
ls -l
B 14:55 Rec
cd RedFlix

src — Python • Python manage.py runserver 8005 —
amankhoja@cometnet-10-21-1-149 RedFlix % source bin/activate
(RedFlix) amankhoja@cometnet-10-21-1-149 RedFlix % cd src
(RedFlix) amankhoja@cometnet-10-21-1-149 src % python3 manage.py runserver 8005
Performing system checks...
System check identified no issues (0 silenced).
March 13, 2020 - 20:08:27
Django version 2.0.7, using settings 'Redflix.settings'
Starting development server at http://127.0.0.1:8005/
Quit the server with CONTROL-C.
[13/Mar/2020 20:08:33] "GET / HTTP/1.1" 200 9281
Not Found: /w3images/onepage_restaurant.jpg
[13/Mar/2020 20:08:33] "GET /w3images/onepage_restaurant.jpg HTTP/1.1" 404 2264
Not Found: /w3images/chef.jpg
[13/Mar/2020 20:08:33] "GET /w3images/chef.jpg HTTP/1.1" 404 2222
Not Found: /w3images/map.jpg
[13/Mar/2020 20:08:33] "GET /w3images/map.jpg HTTP/1.1" 404 2219
Not Found: /favicon.ico
[13/Mar/2020 20:08:33] "GET /favicon.ico HTTP/1.1" 404 2204
```

5.2b Backend Server

5.3. Dynamic model



5.3a Sequence Diagram

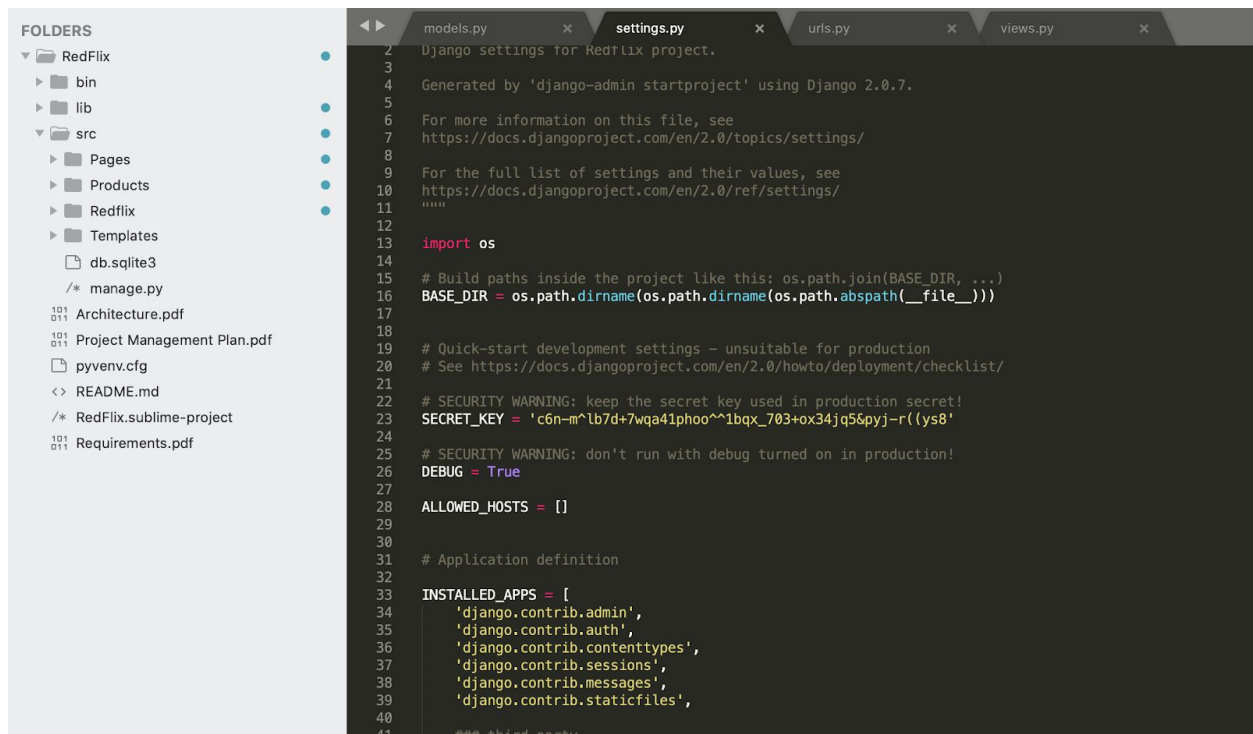
First the web user will request video from the CDN. The CDN will request our server for the videos the user wants. Once the request is received, our server will fetch the videos from the Server data like Hulu or Youtube. The user has the option to create their own library, which is done by the method `prioritievideo()`, which will save the file to our server and they can delete it whenever they want. The user must be able to login, which the information will be saved to the server and also the option to register for the service, where the user can select the type of plan they want.

5.4. Rationale for your detailed design model

User Interface and Front End design have been picked due to the current demands of the industry. For this design we aimed for an implementation that would be easy for the user to manage through in a spaced out manner. Everything that the user

could want is all on the navigation bar, all they need to do is click on it and they will be redirected to the portion of the website that they like.

Text Editor- Backend



5.4a

We have set up a web based app using Django version 2.0.7. For testing purposes we will be using the sqlite3 database system and will slowly migrate to Amazon AWS server. We will be creating a different class for each component like Pages for different pages supported by our web application.

5.5. Traceability from requirements to detailed design model

FUNCTIONAL REQUIREMENTS:

- A user shall be able to log in
- A user shall be able to click a Support tab despite membership status
- A user shall be able to click an About Us tab despite membership status
- A user shall be able to register via facebook, google, or preferred email
- A user shall be able to browse available content
- A user shall be able to search available content

A user shall be able to select videos for playback on six screens simultaneously
A user shall be able to expand one of six selected videos
A user shall be able to switch expansion of either six screens
A user shall be able to stream videos
A user shall be able to fast-forward videos
A user shall be able to pause videos
A user shall be able to rewind videos
A user shall be able to scrub videos
A user shall be able to adjust stream volume
A user shall be able to access the home screen at any point
A user shall be able to log out
The system shall redirect the user to the Watch screen when the user clicks on a platform in the Main screen
A user shall be able to contact customer service in the Support tab via email/phone number
The system shall display FAQs in the Support tab
A user shall discover company details in the About Us tab
The system shall exit the full screen mode via the esc key

NON-FUNCTIONAL REQUIREMENTS:

Security

The system shall be secure with 99.9% certainty
The system shall accomplish this by outsourcing all security related implementations and problems to another software company
The user data will be stored securely on a cloud based system.
The cloud based storage should provide about 200TB worth of storage.
The system shall require users to use a password that is 8-20 characters long
The system shall require at least 1 uppercase letter, 1 lowercase letter, and 1 number
Access rights authorizing on who is allowed to watch both live as well as on demand streams has to be possible on per stream basis (AD group based or individual AD members)

Availability

The system shall be available 24/7
If a user forgets their login credentials the system will send a reset password email within 4 seconds to the accounts email address
The streaming system must be designed and delivered in high availability configuration with built in redundancy to ensure smooth operation even when one component fails.

Scalability

The system shall be able to reach over 100,000 users

For connection of up to 500 external users, Cloud based services are allowed but all internal connections have to be streamed from on premise systems.

The systems bandwidth shall have a minimum bandwidth of 65 Mbps

This shall be best achieved if the user has a fiber optics connection or ethernet cable.

Reliability

When the user registers and sets up a account with the system there private information secure with us

There information will be available to them upon the request of the user request

The user must be able to log into the system with their credentials (email and password)

Usability

Front end

The system shall clearly display a button/click option for the home page

The system shall showcase a drag-and-drop feature when ordering currently playing media

The system shall clearly display a search feature

Back end

The system will use AWS to store customer data in a secure manner that is easily accessible to the user upon request

Languages that the system will use for backend will be Javascript

6. TestPlan

6.1. Requirements/specifications-based system level test cases

Test Case 1 (Table 2)

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:	<ol style="list-style-type: none"> 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 3)

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:	<ol style="list-style-type: none"> 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 4)

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:	<ol style="list-style-type: none"> 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 5)

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:	<ol style="list-style-type: none"> 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 6)

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:	<ol style="list-style-type: none"> 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 7)

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:	<ol style="list-style-type: none"> 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 8)

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:	<ol style="list-style-type: none"> 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 9)

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:	<ol style="list-style-type: none"> 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 10)

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:	<ol style="list-style-type: none"> 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 11)

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:	<ol style="list-style-type: none"> 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 to the library page.
Expected Results:	The user will have list of videos in the library section

6.2. Traceability of test cases to use cases

(Table 12)

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					

6.3. Techniques used 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.

6.4. Assessment of the goodness of your test suite
(Which metrics were used for such assessment?)

Login

To login please enter your email and password

Please fill out this field.

ENTER PLATFORM

Registration

To register for our service please fill out the information below

David Mathew

justinTime@gmail.com

Credit Card

Please fill out this field.

Your suggestions to improve

123456789

GIVE ME MY SUBSCRIPTION

7. Acknowledgment References

J. Kile, “Agile Software Development Quality Assurance,” *Agile Software Development Quality Assurance*, pp. 186–205, May 2019.

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].

“Stream TV and Movies Live and Online,” *Hulu*. [Online]. Available:

<https://www.hulu.com/>.

“Unlimited movies, TV shows, and more.,” *Netflix*. [Online]. Available:

<https://www.netflix.com/>.

“YouTube Premium,” *YouTube*. [Online]. Available: <https://www.youtube.com/premium>.

“IEEE Code of Ethics,” *IEEE*. [Online]. Available:

<https://www.ieee.org/about/corporate/governance/p7-8.html>.

“Software Engineering Code,” *ACM Ethics*, 19-Dec-2018. [Online]. Available:

<https://ethics.acm.org/code-of-ethics/software-engineering-code/>.