Linnéuniversitetet
Kalmar Växjö

OVS - Online Streaming service

Requirements Document

Author: LEONARDO PEDRO

Semester: Spring 2020

Email lr222qp@student.lnu.se

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to give an organized and detailed description of the requirements for the online video streaming service named EyeStream! This document is designed to act as a guide for the developer's team and ensure a mutual understanding of the software's functionality and contents between co-workers and is also used as a software validation for the users which includes all known technologies, pitfalls and necessary structure to develop EyeStream!. This document should include all the necessary information to start the development of the OVS system and explain systems constraints, interface and interactions with external applications.

1.2 Scope

The scope of this product is to present and deliver a good database of tv-shows/series/movies as well as an attractive and user-friendly interface so that the user can easily use it and enjoy the experience of online streaming. EyeStream! will have a membership fee that every user will have to pay (subscribe) to access the tv-shows/series/movies database. The user will also have the feature of accessing the list of movies/tv show/series previously watched, the recommend list depending on his previously watched ones so it's beneficial for both company and users.

1.3 Stakeholders

This section will list all stakeholders as well as a short description of each stakeholder in this project.

Stakeholders List

• **Software Developer:** A software engineer that works with developing, testing and maintaining software.

• **End-User:** Any person with the desire of watching a movie/serie/tv-show that seeks for a good experience of entertainment.

• **Examinator:** An experienced software developer that will monitor and evaluate the project.

• **Investors:** The buyers of the system with a financial interest.

1.4 Overview

This requirements document is destined for the development of an Online Streaming Service called EyeStream!. In this document all the requirements along with all the definitions related to OVS can be found in section 1. all the requirements and analysis of the product can be found in section 2 followed with section 3 that explains all the functional and non-functional requirements.

1.5 Definitions, Acronyms and Abbreviations

MyList - MyList is the location folder where the user can save their videos to watch it later.

Subscription - A contract, paid in advance that will return something, give something to the user.

2 General Description

2.1 Product Prespective

EyeStream is a new online streaming service that allows the user to watch all the movies and series available on their platform. Nowadays, everybody in our society demands a good entertainment source and that's exactly where EyeStream! strikes. Recent studies show that regular people are looking for movies and series online, spending a lot of unwanted time searching websites to try and find the movie/series they want to watch, either on video streams or files that download the video that can become really obnoxious to deal with. Our system aims to make a smooth and enjoyable experience for the user while browsing through the app, providing with high quality and fast response from the OVS.

2.2 Product Functions

EyeStream! software should support computerized network over the internet, providing a variety of categories, all arranged by order with the option of filtering so the user can select only what he wants. The website/app will have 3 main categories, Movies, Series and TV-shows where everything can be found, also having some subcategories included in at least one of the main ones, Anime, Movies Genres, etc..The user interface will have a search bar where the user can type the desired movie/series/tv-show name and be presented with all the suggested results related to his search bar input. The website also has some secondary features where it allows the user to download, watch online, save offline and directly add the video into “MyList” to watch them later. Once the user clicks on the selected video the buffering starts and the video starts playing, this way in runs at the same time providing the user with a good broadcasting experience regarding speed/quality of the video.Figure 1. Use case for Play Video

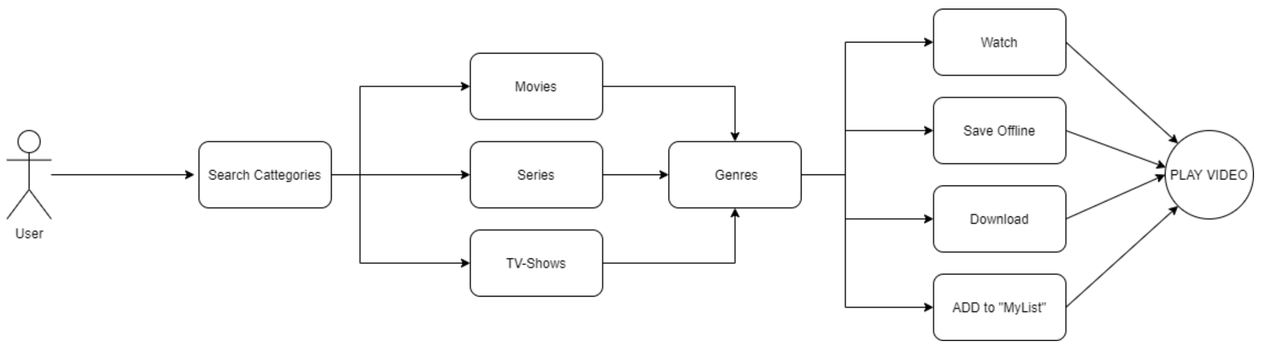

Figure 1. Use case for Play Video

2.3 User Characteristics

The user will be subscribed to a membership fee which allows him to watch anything he wants, since the user is paying for such streaming service and will most likely spend time browsing through the website, EyeStream aims to present a good looking interface with a user-friendly broadcasting service that can deliver any video with the best quality/speed.

EyeStream! will have two levels of user. The first will be the general user which uses the website as a platform to gather all the movies and shows that they look for. The normal customer will be searching for a movie/series/tv-show on a daily bases so he will be systematically scrolling, searching, changing categories, etc.. these users will only interact with the front end of the system. The second level of the user is the developers that will be interacting with both front and back end of the system since they are in charge of making sure everything runs accordingly, possible pitfalls, incompatibilities with technologies, video formats, etc..

2.4 General Constraints

While developing this OVS some constraints should be noted:

- Access large amounts of data with a low internet speed.
- Storage of huge data
- Handling different types of video formats in the database
- Matching the server request to the client.
- Server capacity
- Security

2.5 Assumptions and dependencies

In this project it is assumed that the database for this web will be fully operational, the server capacity is large enough to host multiple-member entries and the internet speed used by the user is enough and sufficient to load all the data from the database/server to the front end without compromising or losing packets.

3. External Interface Requirements

3.1 User Interface

EyeStream! aims to design the website and all the user interfaces in the best way regarding user interests, making sure the user can have an easy and enjoyable experience when searching and browsing through the app. To facilitate the customer experience the interface will consist of one screen/home page where every action takes place avoiding multiple windows/tabs open that will create a disorganized and messy interface. That main screen will contain the company logo, one login button, one sign-in button, one column to present all the categories, one subsection on the column to access MyList and one search bar on top of the homepage like it is shown on Figure 2.

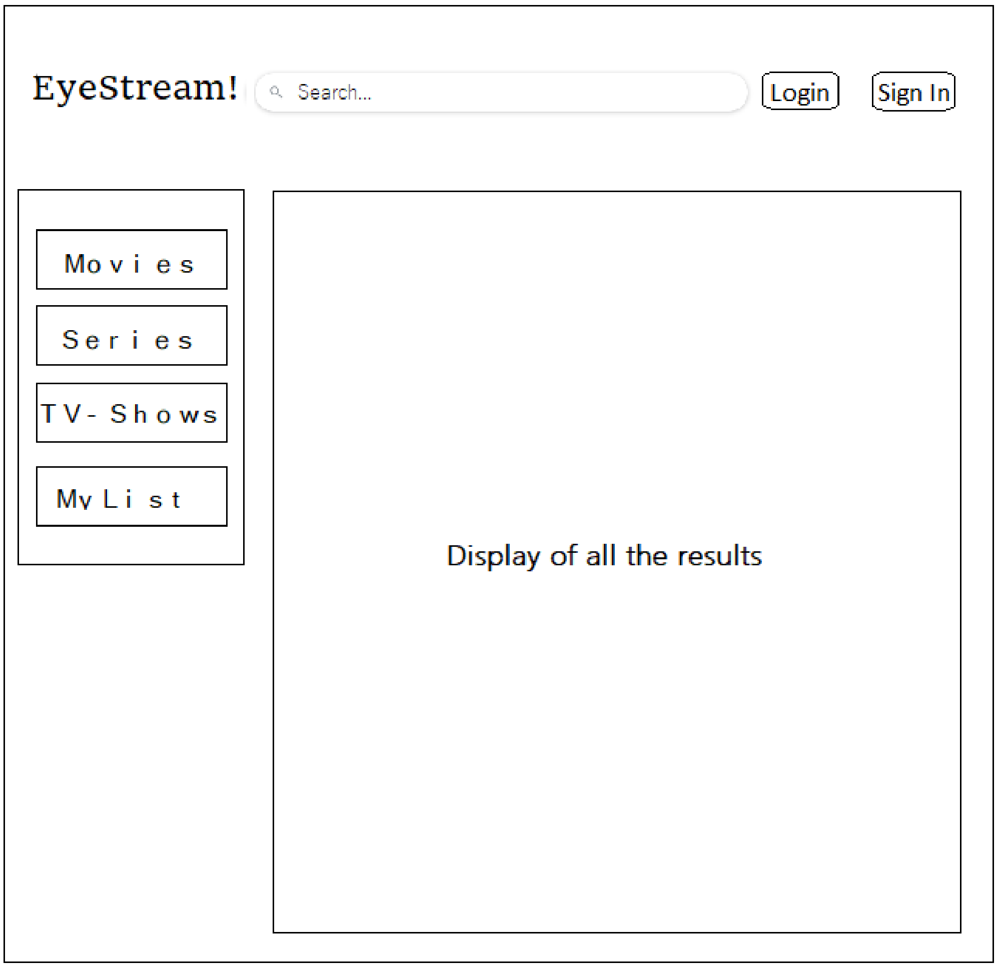

Figure 2. Main User interface

3.2 Hardware Interface

In this OVS service, the external hardware interface will be used to access video streaming sites to provide the user with multiple video streaming sources which they can choose from.

Communication Interface - Languages such as PHP, Java and others will be used to query the different servers that the system is using and send that data back.

Performance Requierments :

- The system must complete the query of data in less than 5 seconds
- The system must send the URL link to the default browser in less than 3 seconds
- The system must sort all the query data in less than 2 seconds

3.2 Software Interface

EyeStream! can be accessed in any operating system with a web browser and a reliable (speed/quality) internet connection since it will be required to load the videos from the existing source. The user will be able to use the website/app on the computer, mobile phone or any other touch device such as tablets, smartphones, smart TVs, etc.. making this OVS service available on every platform.

4. Specific requirements

4.1 Functional Requirements

RC1 - Requirement 1

Description: The system must redirect the user to the homepage after using the URL link
Input: URL link using the keyboard
Output: The system shows the homepage

RC2 - Requirement 2

Description: The system must be able to login any already registered member
Input: Username/ Email and password using keyboard
Process: Store the information in database and check if the member already exists, send a verification code so the log in can be executed with no problems.
Output: The system shows a welcome/log in message and redirects the user to the homepage.

RC3 - Requirement 3

Description: The system must find a specific video when the customer uses the search bar
Input: Type the video name using the keyboard don the search bar
Process: System checks in database and match query with data.
Output: Display all the results relating the input data to the user

RC4 - Requirement 4

Description: Add a video to MyList
Input: Click on the button to add video to MyList
Process: System adds the video to the member's list (MyList)
Output: Display MyList page together with all the previously listed videos

RC5 - Requirement 5

Description: Checking for video details
Input: Video name
Process: Check database for the movie and get all the video details
Output: Display video details such as full name, release date, length and size

4.2 Non-Functional Requirements

4.2.1 Performance

Performance Requirements :

- ➢ If user does not respond to the subscription plan for the amount of time that it was issued to him, then the subscription will be automatically cancelled
- ➢ If video is not available the system should show video not found message

4.2.2 Security

The main security concern is the user, hence the account (log/sign in) mechanism that should keep hackers away from fishing private data.

Security Requirements :

- ➢ All user information and accessibility is censured and protected
- ➢ Every 6 months the system sends a recommendation email to all users advising them to change password.
- ➢ Every user will have only 3 attempts to complete their log in correctly

4.3 Checklist for Requirements Analysis

	RC1	**RC2**	**RC3**	**RC4**	**RC5**
Requirements ambiguity	No	No	No	No	No
Combined Requirements	No	No	No	No	No
Use of non-standard hardware	No	No	No	No	No
Conformance with business goals	Yes	Yes	Yes	Yes	Yes
Premature design	Yes	Yes	Yes	Yes	Yes
Requirements realism	Yes	Yes	Yes	Yes	Yes
Requirements testability	Yes	Yes	Yes	Yes	Yes

4.4 Classification on Faceted Approach

	RC1	RC2	RC3	RC4	RC5
System	Yes	Yes	Yes	Yes	Yes
User Interface	No	Yes	Yes	No	Yes
Database	No	Yes	Yes	Yes	Yes
Communication	Yes	Yes	Yes	No	No

4.5 Systematic Validation

In this section the work of validating the requirements will be presented as a table where each requirement gets marks if they fulfill a point in the checklist.

The checklist includes:

[C1] Completeness: There is not any requirement missing or missing information from the individual requirement description

[C2] Consistency: The description does not include contradictions

[C3] Comprehensibility: The requirement is written in a way that is understandable

[C4] Ambiguity: There are not multiple ways of interpret the requirement

[C5] Structure: This requirement is organized so that it is grouped with other relatable requirements

[C6] Traceability: This requirement can be traced to related requirements

[C7] Standards: The individual requirement conform to defined standards

ID	C1	C2	C3	C4	C5	C6	C7
RC1	✓	✓	✓	✓	✓	✓	✓
RC2	X	✓	✓	✓	✓	✓	✓
RC3	✓	✓	✓	✓	✓	✓	✓
RC4	X	✓	✓	✓	✓	✓	✓
RC5	✓	✓	✓	X	✓	✓	✓

4.4 Other Requirements

Functional Requirements:

- The system must register a new member once the membership fee is paid
- The system must send an email when a new user is registered
- The system must notify the user when a new season is released (from the user's movie/series watched list)
- The system must timeout the broadcasting/streaming session from member accounts that are being used at the same time

Non-functional Requirements:

- New registrations should be processed and verified with a latency of 1 minute.
- Emails should be sent with a latency of no greater than 7 hours after a new member is registered
- Notifications should be sent with a latency of no greater than 12 hours when a new season comes out
- Timeouts should be handled with a latency of 10 minutes after both users started sharing the same account and should last until one of them signs off the account

5 Test Cases

TC1.Register Member
Use Case Tested: The system must register a new member once the membership fee is paid ***Short Description:*** This test case is testing if the system registers a new user as member after paying the entry membership fee
TC2.Email Verification
Use Case Tested: The system must send an email when a new user is registered ***Type of testing:*** Manual Testing ***Short Description:*** This test case is testing if the system sends the verification email to all users that register as new members. ***Test steps for both TC1, TC2*** ➢ Open any web browser and search for the web site (www.*******.com); ➢ Go to the register section and input all the data; ➢ Complete the payment successfully ➢ Verify verification email ➢ Verify that you have access to any movie/series/tv-shows

TC3.Member Authenticator
Use Case Tested: The system must authenticate the user as a member when they login ***Type of testing:*** Manual Testing ***Short Description:*** This test case is testing if the system authenticates and verifies that the existing member logging in is authenticated and is a legit user ***Test steps:*** ➢ Go to the website; ➢ Login as a member and input all the correct input; ➢ Check database and verify if the member was authenticated;

TC3.Member Authenticator
Use Case Tested: The system must authenticate the user as a member when they login ***Type of testing:*** Manual Testing ***Short Description:*** This test case is testing if the system authenticates and verifies that the existing member logging in is authenticated and is a legit user ***Test steps:*** ➢ Go to the website; ➢ Login as a member and input all the correct input; ➢ Check database and verify if the member was authenticated

TC4.Recommended Display
Use Case Tested: The system must display the recommended movies/series/tv-shows when a member log in ***Type of testing:*** Manual Testing ***Short Description:*** This test case is testing if the system displays the recommended set of movies after logging in ***Test steps:*** ➢ Go to the website; ➢ Login as a member and input all the correct input; ➢ Check the set of movies displayed on the front page after log in ➢ Check database and check the array of the recommended movies in display ➢ If they were the same set of movies in display, the test is completed