Software Requirements Specification

Version 1.0

StreamTrace Web Application

Theme: StreamTracker
Category: Web Application Development

Contents

1.1 Background and Necessity for a Web Application

Streaming services have become increasingly popular and have transformed the way we consume media. Streaming is an immediate and continuous method of accessing content from the Internet. It has become the predominant way for people to experience music and videos.

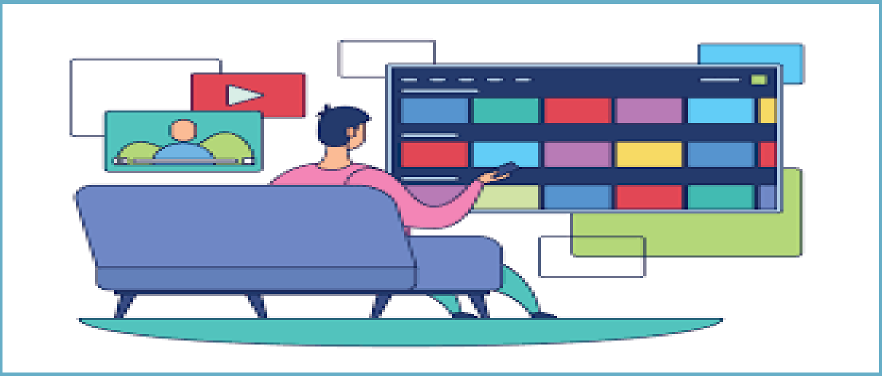
With the increasing number of streaming services available, users often subscribe to multiple platforms. Keeping track of various subscriptions, managing billing cycles, and organizing content across different services can become cumbersome. A Web Application can offer a centralized platform where users can manage and track their subscriptions, view upcoming payments, and maintain a comprehensive overview of their streaming services in one place.

1.2 Proposed Solution

The proposed solution is a Web Application called '**StreamTrace**' that allows users to track and manage their streaming services and subscriptions in one centralized platform. The application should provide features such as adding/modifying/deleting streaming providers, marking favorite shows, setting reminders for upcoming streaming payments, reminders for upcoming shows, searching, sorting, and filtering functionalities.

1.3 Purpose of the Document

The purpose of this document is to provide a comprehensive overview of the Web application named **StreamTrace**. This document explains the purpose and features of **StreamTrace**, the interfaces of the Web application, what the application will do, and the constraints under which it must operate. This document is intended for both stakeholders and developers of the Web application.



1.4 Scope of the Project

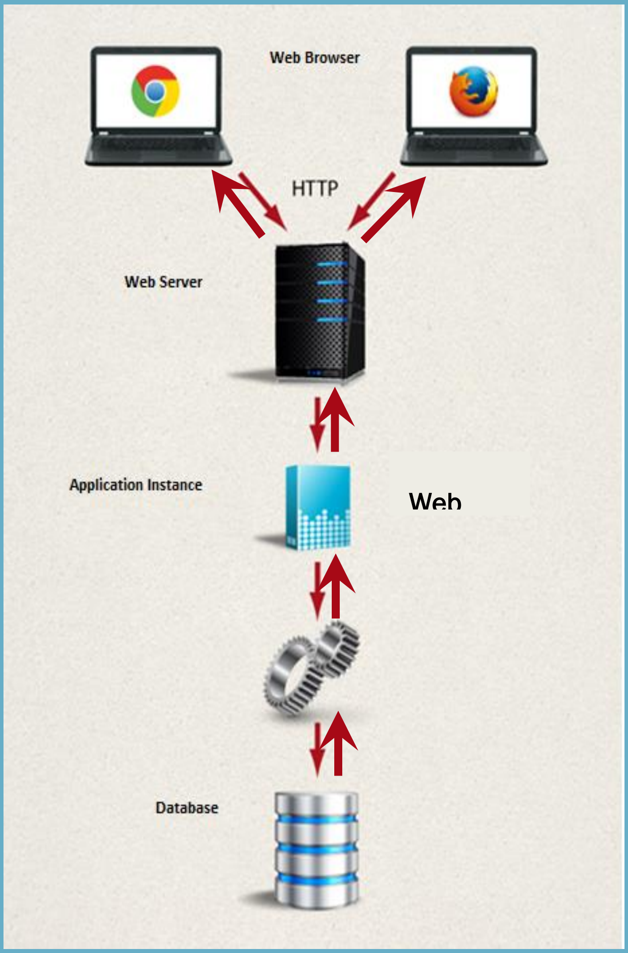
This Web application will be a responsive and visually appealing one to be used by individuals. The scope of **StreamTrace** Web application can encompass various functionalities and features. It will be designed to aid users in maintaining various subscriptions and related information as subscription renewal date, subscription payments, favorite shows, and so on.

However, the application will not have any feature/functionality for implementing or authenticating payment or content streaming. These actions are beyond the scope of this application.

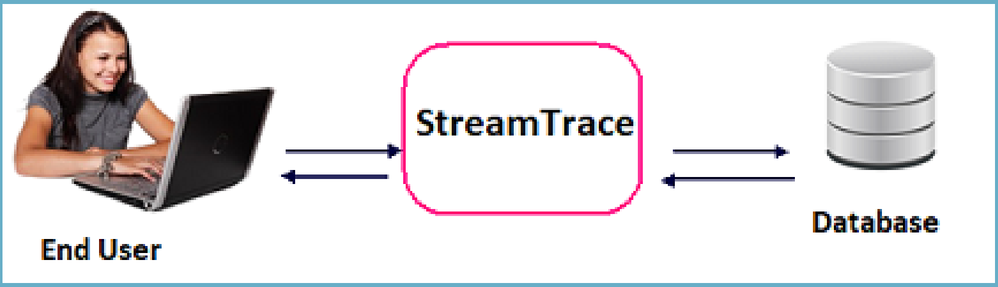
1.5 Constraints

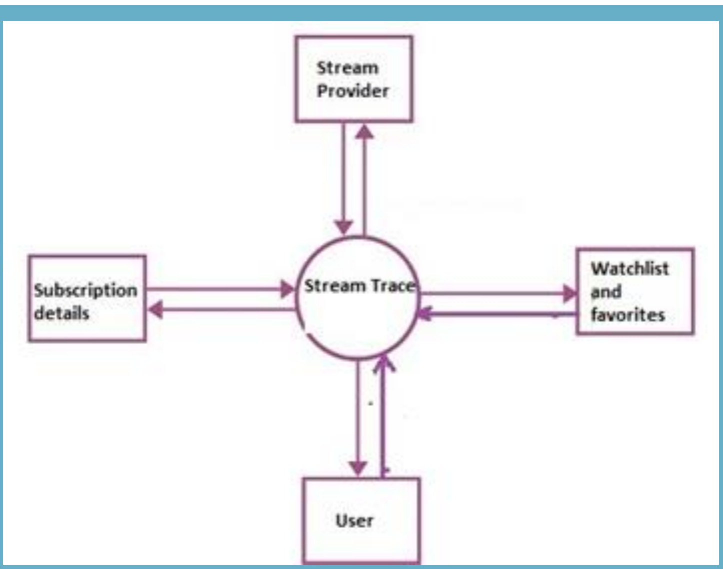
Streaming services typically handle a large number of concurrent users who expect high-quality video playback and seamless streaming experiences. Ensuring scalability and performance constraints involves optimizing the infrastructure to handle increased traffic, reducing buffering times, and delivering a consistent streaming experience across different devices and network conditions. These features are beyond the scope of this Web application.

Architecture Diagram for StreamTrace



Flow diagrams for Stream Trace





1.6 Functional Requirements

Following are major features expected from the Web application, **StreamTrace**:

- **Home Page:** The home page should typically showcase a dashboard that will have the previously watched show/subscription details. It should also provide a welcoming introduction to the Web application.

- **Account Registration:** The registration option shall allow users to create secure accounts. It will enable new users to register themselves with **StreamTrace**. At the time of registration, users must provide Name, Email ID, Contact Number, and Username, and then, configure their Password.

  Appropriate error-checking must be done on the fields of the form to ensure correct data. For example, email id can be checked to see if it is of appropriate format. (Hint: Use client-side validation).

- **Login**: It will allow successfully registered users to login to the **StreamTrace** application and access various features of the application through menus or sidebars. A profile can include detailed information about the user, such as age, and so on.

- **Settings:** Users will be able to manage their accounts by using sub options such as Create, Update, and Delete Profile. Users can also add addresses using this option.

- **Streaming Service Provider Management:** User should be able to add/update and delete various service providers along with details such as Name, Logo, and so on**.**

- **Subscription Management:** This will allow users to add, edit, or cancel subscriptions to various streaming services. Each subscription will have fields such as Name, Logo, Subscription URL, Renewal date, Amount, and so on. This option should also provide reminders for upcoming subscription payments or renewal dates.

- **Recommendations:** This option will implement personalized content recommendation based on user preferences, viewing history, and ratings.

- **Watchlist and Favorites Management:** The application should enable users to create and manage a Watchlist of movies, TV shows, or music albums from different streaming services. It should allow users to mark favorites and track already watched content.

- **Search/Sort/Filter:** Users can sort data, search or filter a specific show/movie, and so on.

- **Contact Us:** This menu option should display Email id, address, and contact number of the organization who is developing the application.

- **Feedback**: This option should display a feedback form using which users can provide their feedback.

- **Sitemap:** To understand the flow of **StreamTrace** Web application, you will have to create a Sitemap and add it to the home page of your application.

Note: Boilerplate or readymade HTML template can be used, provided it is only for design aspect and not for implementing application functionality.

1.7 Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the Web application.

The application should be:

- **Safe to use**: The application should not result in any malicious downloads or unnecessary file downloads.
- **Accessible**: The application should have clear and legible fonts, user-interface elements, and navigation elements.
- **User-friendly**: The application should be easy to navigate with clear menus and other elements and easy to understand.
- **Operability**: The application should operate in a reliably efficient manner.
- **Performance**: The application should demonstrate high value of performance through speed and throughput. In simple terms, the application should be fast to load and page redirection should be smooth.
- **Security**: The application should implement adequate security measures such as authentication. For example, only registered users can access certain features.
- **Scalability**: The application architecture and infrastructure should be designed to handle increasing user traffic, data storage, and feature expansions.
- **Capacity**: The application should support large number of users.
- **Availability**: The application should be available 24/7 with minimum downtime.
- **Compatibility**: The application should be compatible with latest browsers.

These are the bare minimum expectations from the project. <u>It is a must to implement the functional and non-functional requirements given in this SRS.</u> Once they are complete, you can use your own creativity and imagination to add more features if required.

1.8 Interface Requirements

1.8.1 Hardware

Intel Core i5 Processor or higher
8 GB RAM or higher
Color SVGA
500 GB Hard Disk space
Mouse
Keyboard

1.8.2 Software

Technologies that can be used:

- <u>Frontend</u>:

  HTML5, CSS3, Bootstrap, JavaScript, Figma Toolkit, jQuery, AngularJS/Angular 9/ReactJS, and XML

- <u>Client and Server</u>:

  Java 9 or higher, Java EE 7 or higher/Jakarta EE 9 or higher, with Apache NetBeans IDE/Eclipse latest version, Apache Tomcat 10.0 or higher, GlassFish 6.0 or higher, and related libraries

  OR C# 7.0 with Visual Studio IDE 2019 or higher, ASP.NET MVC and Core, and related libraries

  OR PHP 7.0 or higher version with Laravel Framework Homestead (optional)

  OR Python 3.0 or higher version with PyCharm IDE, Django 4.0.2 or higher/Flask framework

  For hosting (optional):
  XAMPP latest version

- <u>Data Store</u>:

  MySQL 5.7 or higher/SQL Server 2016 or higher

Database Design

Data Dictionary: Users, Streaming Service Provider, Subscriptions, and so on. Based on the given specifications, you will define suitable entities, attributes for these entities, and identify relationships between the entities.

For example, some entities along with their attributes can be identified as follows:

Users
• User_Id (Primary Key)
• Username
• Email
• Password

Streaming Service Provider
• Service_Id (Primary Key)
• Service Name
• Logo/Icon

Subscription
• Subscription_Id (Primary Key)
• User_Id (Foreign Key referencing Users)
• Service_Id (Foreign Key referencing Streaming Service Provider)
• Subscription Start Date
• Subscription End Date
• Billing Amount

Similarly, you can define other entities and also relationships between entities and methods representing activities on the entities.

Note: These are just examples, you do not have to adhere to these structures and can design your own table structure with more or less columns.

1.9 Project Deliverables

You will design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design specifications
- Diagrams such as flowcharts for various activities, Data Flow Diagrams, and so on
- Database Design (database scripts to be provided)
- Source Code
- Installation guide
- User Manual along with test data and login credentials

Documentation is considered as a very important part of the project. Ensure that documentation is complete and comprehensive. The consolidated project will be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and SQL scripts files (.sql) containing database and table definitions. If local servers such as XAMPP are used to test the project, include screenshots of working pages. Include port settings and other details in ReadMe file.

In addition, you must submit a video clip showing the actual working of the Web application.

Over and above the given specifications, you can apply your creativity and logic to improve the application.

*~~~ End of Document ~~~*