

Software Requirements Specification For CompareStream

2/16/18
Version 1

Prepared by:
Hamza Chaudhry, Adrian Williams, Kristopher Berk-
land, Joel Brandom, Jacob Brandom, Min Xiong

Table of Contents

| | |
|---|----------|
| REVISION HISTORY | 3 |
| 1. INTRODUCTION | 4 |
| 1.1. Overview..... | 4 |
| 1.2. Goals and Objectives..... | 4 |
| 1.3. Scope | 5 |
| 1.4. Definitions | 5 |
| 1.5. Document Conventions | 6 |
| 1.6. Assumptions | 6 |
| 2. GENERAL DESIGN CONSTRAINTS..... | 6 |
| 2.1. Product Environment..... | 6 |
| 2.2. User Characteristics | 7 |
| 2.3. Mandated Constraints | 7 |
| 3. NONFUNCTIONAL REQUIREMENTS..... | 7 |
| 3.1. Usability Requirements..... | 7 |
| 3.2. Operational Requirements | 7 |
| 3.3. Performance Requirements..... | 7 |
| 3.4. Security Requirements..... | 7 |
| 3.5. Other Requirements..... | 8 |
| 3.6. Interface..... | 8 |
| 4. SYSTEM FEATURES | 8 |
| 4.1. Feature: Log in / Check settings and List of Favorite Shows..... | 8 |
| 4.1.1. Description and Priority | 8 |
| 4.1.2. Use Case: | 9 |
| 4.2. Feature: Check Streaming Deals | 9 |
| 4.2.1. Description and Priority | 9 |
| 4.2.2. Use Case: | 9 |
| 4.3. Feature: Check Invalid Login Attempts | 10 |
| 4.3.1. Description and Priority: | 10 |
| 4.3.2. Use case: | 10 |
| 4.4. Feature: Add TV Show to Services..... | 11 |
| 4.4.1. Description and Priority:..... | 11 |
| 4.4.2. Use case: | 11 |

| | |
|--|-----------|
| 4.5. Feature: Add Streaming Service..... | 11 |
| 4.5.1. Description and Priority:..... | 11 |
| 4.5.2. Use case: | 12 |
| 4.6. Feature: Edit/Remove Streaming Services..... | 12 |
| 4.6.1. Description and Priority: | 12 |
| 4.6.2. Use case: | 12 |
| 4.7. Feature: Report Problems | 13 |
| 4.7.1. Description and Priority: | 13 |
| 4.7.2. Use case: | 14 |
| 4.8. Feature: Look Up Services and Which TV Shows They Offer..... | 14 |
| 4.8.1. Description and Priority: | 14 |
| 4.8.2. Use case: | 14 |
| 4.9. Feature: View Statistics..... | 15 |
| 4.9.1. Description and Priority: | 15 |
| 4.9.2. Use case: | 15 |
| 4.10. Feature: View Reports..... | 16 |
| 4.10.1. Description and Priority: | 16 |
| 4.10.2. Use case: | 16 |
| 4.11. Optional Feature: Anonymous User Sign Up..... | 16 |
| 4.11.1. Description and Priority:..... | 16 |
| 4.11.2. Use case: | 17 |

Revision History

| Version | Date | Name | Description |
|---------|---------|----------------|------------------|
| 1 | 2/16/18 | Hamza Chaudhry | Initial Document |
| | | | |

1. Introduction

1.1. Overview

The CompareStream application will be a web application available to all web users. The application will provide users with the ability to enter in their favorite shows and using this it will help decide which package has the networks that provide the best deal for all new episodes of that series. If multiple shows are entered the user will be able to rank them, in accordance to their importance, so as to ensure the best package is provided. Users can also create an account to save and alter their settings as they wish.

This application also provides admin users the ability to work with user data. It also allows for admins to manipulate and determine streaming services, prices, networks and shows that are on the networks.

This document provides information on the requirements for the CompareStream software application. Project goals, scope and definitions are given in the introduction. Design constraints and application environment are described in the following section. Non-functional requirements are outlined for later verification. Functional requirements are given to show the system features and expected user interaction.

Project constraints will be included in separate documentation. The Software Project Management Plan will give specifics on project budget and schedule. A separate Test Plan document will address test specifications and procedures.

1.2. Goals and Objectives

The three main goals of the CompareStream:

1. Provide a simple mechanism for users to compare TV packages of multiple streaming services based on what shows they like the most and what networks provide those shows.
2. Allow user to rank shows in accordance to which ones are more important to being included.
3. Provide users with the best offer after considering entered data.
4. Allow user to log in with their own account and save and store settings.
5. Implement security measures for account safety.
6. Allow anonymous users to look for shows and what services offer those on which networks.
7. Allow all users to report problems as necessary.
8. Allow admins to work with specific parts of user data.

9. Allow admins to manipulate data for streaming services, prices, networks and shows that are provided on those networks.
10. Provide admins with reports on invalid login attempts, popular shows, most commonly chosen service, etc.
11. Work in a relatively intuitive manner that does not require too much from the user.

1.3. Scope

The CompareStream app will allow have 3 types of users. An admin user will be allowed to manipulate provided data as well as work with user data. Anonymous users can use the app for its functionality and leave as they wish. A logged in user can save and alter settings for ease of use and remembrance. All users will be able to report problems with shows or networks.

1.4. Definitions

This section defines potentially unfamiliar or ambiguous words, acronyms and abbreviations. If terms such as “shall”, “should” and “may” are used to indicate importance the meaning of these terms should be defined here.

Example:

Use case – describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.

Scenario – one path through a use case

Actor – user or other software system that receives value from a use case.

Role – category of users that share similar characteristics.

Product – what is being described here; the software system specified in this document.

Project – activities that will lead to the production of the product described here. Project issues are described in a separate project plan.

Shall – adverb used to indicate importance; indicates the requirement is mandatory. “Must” and “will” are synonyms for “shall”.

Should – adverb used to indicate importance; indicates the requirement is desired but not mandatory.

May – adverb used to indicate an option. For example, “The system may be taken offline for up to one hour every evening for maintenance.” Not used to express a requirement, but rather to specifically allow an option.

Controls – the individual elements of a user interface such as buttons and check boxes.

User — a registered user on the CompareStream website

Anonymous user — an unregistered user on the CompareStream website

Admin — a user and owner/moderator of the CompareStream website

1.5. Document Conventions

This section describes presentation conventions use in the document.

Example:

Portions of this document that are incomplete will be marked with TBD. Each TBD item will have an owner and estimated date for resolving the issue.

1.6. Assumptions

In this section list any assumptions on which the requirements, as they are described here, depend. Assumptions are conditions, usually outside the control of the performing organization, that are taken for granted.

Be careful to only document assumptions that are outside the control of the performing organization. For example, the following is not a valid assumption for the requirements document: “We assume that all requirements will be documented.” You might be *assuming* this but there is no point in documenting it as an assumption here because it is something that is primary within the scope and control of the performing organization. The purpose of this section is to document assumptions that are outside the control of the performing organization—conditions that should be noted because someone will be taking responsibility for ensuring they hold.

A distinction can also be made between assumptions that pertain to the requirements and those that pertain to the project as a whole. The software project management plan is the most logical place to document assumptions that pertain to the project as a whole.

Example:

It is assumed that the client has an ODBC compliant database installed and this database will be accessible from the machine where the system will run.

It is assumed that the web hosting ISP will allow server-side scripts to access the file system.

2. General Design Constraints

2.1. Product Environment

The CompareStream product will be an ASP.NET web application for any platform with a web browser. This application will use the .NET framework for the website, which will interface with a proxy server. The proxy server will interface with the database. We will

connect to an SQL database with an ADO.NET interface to store and analyze data for the user and customer.

2.2. User Characteristics

Three types of Users: admins, a logged in user, and an anonymous user. Admins can edit/add/delete the streaming services, prices and networks, also can work with the users' data; A logged in user can save and store their settings; anonymous user can browse services and the TV shows offered by them. All users can report problems.

2.3. Mandated Constraints

The application will run on any platform with a web browser. This platform, and C#/ASP.NET was chosen by the customer.

3. Nonfunctional Requirements

3.1. Usability Requirements

Users who had difficulty knowing which streaming service was best, should not need to work too hard to use the streaming comparison site. Otherwise, it defeats the purpose of making their choices easier. Therefore, the system should be easy to learn.

In addition the system should make the users want to register and use the site frequently. Thus, the system must be fun to use.

The system shall be easy to learn.

5 out of 6 users shall report that the system was easy to learn.

The system shall be fun to use.

5 out of 6 users shall report that the system was fun to use.

3.2. Operational Requirements

The system shall run on a .net framework web application.

3.3. Performance Requirements

There are none yet.

3.4. Security Requirements

The System may allow for 2-factor authentication.

The System shall lockout users/admins attempting to login more than three times unsuccessfully within 1 minutes for a period of 10 minutes.

Anonymous users shall be restricted to browsing shows and services.

Admin shall be able to access/edit all data. The admin will also receive reports on invalid login attempts.

Registered users shall be able to manage account details, and create lists of favorite shows.

3.5. Other Requirements

The system should be able to handle cases where shows are not available, either due to later releases or diminished availability of previous seasons.

Combining streaming offers should be possible.

Anonymous users reports should not be spamnable.

3.6. Interface

The System interface should be easy to learn and use.

5 out of 6 users should report less than 1 critical problem with navigation.

4. System Features

4.1. Feature: Log in / Check settings and List of Favorite Shows

4.1.1. Description and Priority

User needs to be able to log in to be able to view their settings and their list of favorite tv shows if they've created it.

Actors: Users, Admins

Cost:

Risk:

Value: high

4.1.2. Use Case:

Basic Path

1. User goes to CompareStream website
2. User clicks log in
3. System prompts user to log in
4. User enters correct email and password and click login
5. System displays user settings, favorite tv shows (if available), administrative options (if user is an admin), and option to compare deals
6. User clicks logout.
7. System exits.

Alternate Path

1. User goes to CompareStream website
2. User clicks log in
3. System prompts user to log in
4. User enters incorrect email and/or password and click login
5. System displays error message: "Invalid email and/or password. Or you may have exceeded the number of incorrect login attempts."
6. System locks user out for a day if they failed logging in 5 times within 30 minutes.
7. User may choose to login again, returning to 1. or exit.
8. System exits. Additional Requirements

Include in this section additional functional and non-functional requirements not specified in the use case(s) above.

4.2. Feature: Check Streaming Deals

4.2.1. Description and Priority

Users will be able to select their favorite tv shows, and then based on that information, the system will display the best deals according to their preferences.

Actors: Admins, users

Cost: high

Risk: medium

Value: high

4.2.2. Use Case:

Basic Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays favorite tv show selection
3. User edits their choices of favorite tv shows
4. User selects view deals.
5. System displays best deals according to the users selected tv shows with option to save settings or return home.
6. User clicks return home.

Alternative Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays favorite tv show selection
3. User edits their choices of favorite tv shows
4. User selects view deals.
5. System displays best deals according to the users selected tv shows with option to save settings or return home.
6. User selects save settings.
7. System saves user's favorite tv shows and selected services and displays success message with option to return home.
8. User selects return home.

4.3. Feature: Check Invalid Login Attempts

4.3.1. Description and Priority:

Users may attempt to log in to another user's account. We want administrators to be able to see if this is happening to better report about user's accounts.

Actors: admins

Cost: low

Risk: high

Value: medium

4.3.2. Use case:

Basic Path

1. User goes to CompareStream website

2. Following Login [Use Login Case, Step 5], System displays administrative options
3. User selects View Invalid Login Attempts
4. System displays invalid login attempts (if any exist) or message saying none exist with option to return home
5. User selects return home.

4.4. Feature: Add TV Show to Services

4.4.1. Description and Priority:

Services may gain TV shows, and we need to reflect that in the system for users to have deals accurately tell them what they will be getting.

Actors: Admins

Cost: medium

Risk: low

Value: high

4.4.2. Use case:

Basic Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays administrative options
3. User selects Add TV Show to Services
4. System prompts user to input new TV show and select which services it belongs to.
5. User submits form.
6. System displays success message with option to return home.
7. User returns home.

4.5. Feature: Add Streaming Service

4.5.1. Description and Priority:

New streaming services may arise in the future, so we need a way to add them to our service to better provide our users with current offers to give them the best deals.

Actors: Admins

Cost: medium

Risk: low

Value: medium

4.5.2. Use case:

Basic Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays administrative options
3. User selects Add Streaming Service
4. System prompts user to input new streaming service and select which shows belong to the service.
5. User submits form.
6. System displays success message with option to return home.
7. User returns home.

4.6. Feature: Edit/Remove Streaming Services

4.6.1. Description and Priority:

Streaming services may lose TV shows or go out of business. To keep our service up to date, we need to be able to edit the services and remove them if necessary.

Actors: Admins

Cost: medium

Risk: medium

Value: medium

4.6.2. Use case:

Basic Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays administrative options
3. User selects Edit/Remove Streaming Services.
4. System displays streaming services available to edit.
5. User selects which service to edit.
6. System displays information and current tv shows belonging to selected service with option to remove service.
7. User edits information or adds/removes tv shows for selected service.
8. User submits changes.
9. System displays success message with option to return home.
10. User returns home.

Alternative Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays administrative options
3. User selects Edit/Remove Streaming Services.
4. System displays streaming services available to edit.
5. User selects which service to edit.
6. System displays information and current tv shows belonging to selected service with option to remove service.
7. User selects remove service.
8. System removes service and displays success message with option to return home.
9. User returns home.

4.7. Feature: Report Problems**4.7.1. Description and Priority:**

Problems may arise in the use of this application, and we want the administrators to be aware. Users will be able to report any problems/bugs they have with the website, so they can be fixed.

Actors: Users, anonymous users, admins.

Cost: medium

Risk: high

Value: medium

4.7.2. Use case:

Basic Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays user settings
3. User selects Report Problem.
4. System displays problem reporting form.
5. User inputs description of problem and submits form.
6. System displays success message with option to return home.
7. User returns home.

4.8. Feature: Look Up Services and Which TV Shows They Offer

4.8.1. Description and Priority:

If a user does not sign in, they can still use a limited version of CompareStream where they can only view available services and the TV shows they offer.

Actors: Anonymous users

Cost: medium

Risk: medium

Value: low

4.8.2. Use case:

Basic Path

1. User goes to CompareStream website
2. User selects Look Up Services/TV Shows
3. System displays Services.
4. User selects service.
5. System displays TV shows offered by selected service with option to return to services.
6. User returns to services.
7. System displays services and option to return home.
8. User returns home.

4.9. Feature: View Statistics

4.9.1. Description and Priority:

Administrators may be curious about what tv shows and services are most popular. They may want to tailor other sections of the site to promote the most popular services to make the site stay as mainstream as possible.

Actors: Admins

Cost: low

Risk: low

Value: low

4.9.2. Use case:

Basic Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays administrative settings.
3. User selects View Popular TV Shows.
4. System displays popular TV shows with option to return home.
5. User returns home.

Alternative Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays administrative settings.
3. User selects View Most Commonly Chosen Services.

4. System displays most commonly chosen services with option to return home.
5. User returns home.

4.10. Feature: View Reports

4.10.1. Description and Priority:

After users submit reports of their problems, administrators will want to be able to view the reports, so they can address all issues. They will need to be able to mark reports as fixed to clear up the list of problems and only view the currently unsolved problems.

Actors: Admins

Cost: medium

Risk: medium

Value: high

4.10.2. Use case:

Basic Path

1. User goes to CompareStream website
2. Following Login [Use Login Case, Step 5], System displays administrative settings.
3. User selects View Reports.
4. System displays reports with option to return home.
5. User selects report(s) to mark as fixed.
6. User returns home.

4.11. Optional Feature: Anonymous User Sign Up

4.11.1. Description and Priority:

Anonymous users may be unhappy with only being able to browse the services and available tv shows. So for them to use the full functionality of the website, they will need to sign up.

Actors: Anonymous user

Cost: medium

Risk: medium

Value: low

4.11.2. Use case:

Basic Path

1. User goes to CompareStream website
2. System displays anonymous user options including sign up
3. User selects sign up
4. User inputs email address and password for their account
5. User submits form.
6. System registers user and logs them in, leading them to the user home page.