**Animation Movies**

**SRS - Software Requirement Specification**

< Version 1. 0>

< February25, 2023>

<Raghda Knaane>

STP - Software Test Plan

TABLE OF CONTENTS

[**Revision History:** 3](#_Toc129692672)

[**Document Approval:** 3](#_Toc129692673)

[**1.** **Introduction**  4](#_Toc129692674)

[**2.** **Purpose**  4](#_Toc129692675)

[**3.**  **Scope**  4](#_Toc129692676)

[**4.** **Overview**  4](#_Toc129692677)

[**5.** **System Architecture**  5](#_Toc129692678)

[**6.** **User Characteristics**  5](#_Toc129692679)

[**6.1 User Management System** 5](#_Toc129692680)

[**6.2 Movie Search and Recommendation System** 5](#_Toc129692681)

[**6.3 Movie Details Page** 5](#_Toc129692682)

[**6.4 Review System**  6](#_Toc129692683)

[**6.5 Watchlist System**  6](#_Toc129692684)

[**6.6 Contact Form**  6](#_Toc129692685)

[**7.Functional Requirements** 6](#_Toc129692686)

[**8.Non-functional Requirements**  7](#_Toc129692687)

[**9.Future Enhancements:** 8](#_Toc129692688)

[**10. Glossary**  8](#_Toc129692689)

[**11.** **Conclusion**  9](#_Toc129692690)

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <25/02/2023> | <1.0> | SRS 1.0 | Raghda Knaane |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **Document Approval:**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |

**SRS - Software Requirement Specification:**

## **Introduction**

The Flask web application is a movie review and recommendation system that allows users to browse, search, and rate movies. The application uses a SQLite database to store movie details and user information.

## **Purpose**

The purpose of this document is to specify the requirements of the Flask web application. This document will serve as a guide for the development team to understand the scope of the project and deliver a high-quality product that meets the needs of the stakeholders.

## **Scope**

The Flask web application is designed to provide a user-friendly interface for movie enthusiasts to explore and review movies. The application allows users to create an account, search for movies, view movie details, rate movies, and add movies to a watchlist. The application also includes a contact form for users to send feedback or ask questions.

## **Overview**

The Flask web application consists of the following components:

* User management system.
* Movie search and recommendation system.
* Movie details page.
* Review system.
* Watchlist system.
* Contact form.

## **System Architecture**

The application will be built using the following technology stack:

-          Python 3.7 or higher

-          Flask 2.0.2

-          Flask\_SQLAlchemy 2.6.0

-          Flask Migrate 3.1.0

-          SQLite

## **User Characteristics**

The target audience for this application is movie enthusiasts and collectors who want to manage their movie collections effectively. Users of the application should have basic computer literacy and internet browsing skills.

## **6.1 User Management System**

The user management system is responsible for managing user accounts. Users can create an account, log in, and log out. The system also ensures that only authenticated users can access certain parts of the application.

## **6.2 Movie Search and Recommendation System**

The movie search and recommendation system are responsible for allowing users to search for movies based on their title. The system also provides movie recommendations based on the user's rating history.

## **6.3 Movie Details Page**

The movie details page displays detailed information about a selected movie, such as its title, description, director, release year, actors, and rating.

## **6.4 Review System**

The review system allows users to write and publish movie reviews. Users can rate movies and provide a written review. Users can also edit or delete their reviews.

## **6.5 Watchlist System**

The watchlist system allows users to add movies to their watchlist. Users can view their watchlist and remove movies from it.

## **6.6 Contact Form**

The contact form allows users to send feedback or ask questions to the site administrator.

## **Functional Requirements**

* User Management System**:**
* Users can create an account by providing their name and password.
* Users can log in using their credentials.
* Users can log out of the system.
* The system ensures that only authenticated users can access certain parts of the application.
* Movie Search and Recommendation System**:**
* Users can search for movies by title.
* The system displays search results based on the user's search query.
* The system provides movie recommendations based on the user's rating history.
* Movie Details Page**:**
* The movie details page displays detailed information about a selected movie, such as its title, description, director, release year, actors, and rating.
* The system displays the average rating for a movie.
* Review System**:**
* Users can write and publish movie reviews.
* Users can rate movies on a scale of 1 to 5.
* Users can edit or delete their reviews.
* Watchlist System**:**
* Users can add movies to their watchlist.
* Users can view their watchlist.
* Users can remove movies from their watchlist.
* Contact Form**:**
* Users can send feedback or ask questions using the contact form.
* The system sends an email notification to the site administrator when a user submits a message.

## **Non-functional Requirements**

* The system should be user-friendly and easy to navigate.
* The system should provide fast response times.
* The system should be secure and protect user data.
* The system should be scalable to handle many users and movies.

## **Future Enhancements**

The application can be enhanced in the following ways:

-          Add user authentication and authorization to protect user data from unauthorized access or modification.

-          Allow users to edit their reviews.

-          Add a feature to recommend movies based on user preferences.

-          Integrate with external APIs to provide more movie information.

-          Improve the application's search functionality.

## **Glossary**

The glossary for the Movies Website project includes:

Glossary

Term Definition

Active Article The document that is tracked by the system; it is a narrative

that is planned to be posted to the public website.

Author Person submitting an article to be reviewed. In case of

multiple authors, this term refers to the principal author,

with whom all communication is made.

Database Collection of all the information monitored by this system.

Editor Person who receives articles, sends articles for review, and

makes final judgments for publications.

Field A cell within a form.

Historical Society Database The existing membership database (also HS database).

Member A member of the Historical Society listed in the HS

database.

Reader Anyone visiting the site to read articles.

Review A written recommendation about the appropriateness of an

article for publication; may include suggestions for

improvement.

Reviewer A person that examines an article and has the ability to

recommend approval of the article for publication or to

request that changes be made in the article.

Software Requirements

Specification

A document that completely describes all of the functions

of a proposed system and the constraints under which it

must operate. For example, this document.

Stakeholder Any person with an interest in the project who is not a

developer.

User Reviewer or Author.

Glossary

Term Definition

Active Article The document that is tracked by the system; it is a narrative

that is planned to be posted to the public website.

Author Person submitting an article to be reviewed. In case of

multiple authors, this term refers to the principal author,

with whom all communication is made.

Database Collection of all the information monitored by this system.

Editor Person who receives articles, sends articles for review, and

makes final judgments for publications.

Field A cell within a form.

Historical Society Database The existing membership database (also HS database).

Member A member of the Historical Society listed in the HS

database.

Reader Anyone visiting the site to read articles.

Review A written recommendation about the appropriateness of an

article for publication; may include suggestions for

improvement.

Reviewer A person that examines an article and has the ability to

recommend approval of the article for publication or to

request that changes be made in the article.

Software Requirements

Specification

A document that completely describes all of the functions

of a proposed system and the constraints under which it

must operate. For example, this document.

Stakeholder Any person with an interest in the project who is not a

developer.

User Reviewer or Author.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Database | Collection of all the information monitored by this system. |
| Review | A written comment and rating about the movie. |
| SRS – Software requirements specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer.  Any person with an interest in the project who is not a developer.  Any person with an interest in the project who is not a developer |
| SQLite | SQLite is a lightweight, embedded relational database management system. |
| Flask | Flask is a web framework, it’s a Python module that lets you develop web applications easily. |
| Homepage | The initial page of the website that displays poster for 10 movies. |
| Info pages: | Pages that display detailed information about a specific movie, including video, images, review bar and description. |

## **Conclusion**

The Flask web application is a movie review and recommendation system that provides a user-friendly interface for movie enthusiasts to explore and review movies. The system is built using Python, Flask, and SQL Alchemy and uses SQLite to store movie details and user information. The application consists of several components, including a user management system, movie search and recommendation system, movie details page, review system, watchlist system, and contact form. The system satisfies functional and non-functional requirements and can be enhanced in the future to meet the evolving needs of the stakeholders.