

Software Requirements Specification
(SRS) Document

AniWatch

02/19/2025

Version 1

Foday Mami, Abel Moran, Joshua Wusu

Project General Description

Our web application is an interactive platform designed for anime and manga enthusiasts to track their watchlists, discover new series, and engage with the community. Users can rate and review anime/manga, participate in discussions, and get personalized recommendations. AniWatch hopes to provide an easy-to-use watchlist tracker for anime and manga. Create a review and rating system to help users discover quality content. Offer a discussion platform where fans can engage in meaningful conversations. Allow content providers (reviewers, critics) to curate lists and interact with users.

Product Features

The AniWatch application is designed to allow anime and manga consumers a way to follow watchlist or recommended readings from others within the community of consumers

Personalized Watchlist Tracking: Users can maintain watchlists of their favorite anime's and manga's. Users can also view others watchlist with the ability to rate, save, and comment on the list

Community Engagement: Users can join discussions, share thoughts, and connect with other fans.

Content Curation: Reviewers and critics can create and share curated watchlists.

Rating and Review System: Users can rate series and leave reviews.

User Authentication: Secure sign-up and login system.

Functional Requirements

Customer (Foday)

- FR0: The application shall allow users to create profiles.
- FR1: Users shall be able to manage and update their profiles.
- FR2: Users shall be able to create and share curated watchlists.
- FR3: The system shall provide recommendations based on user preferences.
- FR4: Users shall be able to track their watchlists.
- FR5: Users shall be able to discover new anime and manga series.
- FR6: Users shall have access to community discussions and engagement features.
- FR7: Users shall be able to review and comment on curated watchlists.
- FR8: User accounts shall be secured through a login and sign-up authentication system.

Provider (Joshua)

- FR0: The application shall allow users to establish profiles to share recommendations and lists.
- FR1: Providers shall be able to create curated watchlists, post recommendations, and host discussions.
- FR2: Providers shall be able to analyze user interaction with curates, watchlists and services provided.
- FR3: Providers can engage with users by responding to reviews as well as moderating discussions.

System-Admin (Abel)

- FR0: Application should allow admins to edit, and delete/ban Customer created profiles
- FR1: Admins and system can delete inappropriate comments.
- FR2: Admins and system can filter profanity and age restricted content.
- FR3: Allows the system to display real-time server status.
- FR4: Allows the system to remote control access to reset and clear server cache.
- FR5: System can log all actions for admins and can viewed by administrators.

Non-Functional Requirements

Customer (Foday)

- NFR0: The system shall be able to load user watchlists in under 3 seconds.
- NFR1: The platform shall support concurrent users without performance degradation.
- NFR2: User authentication and authorization shall be secured
- NFR3: Anime and Manga lookup will load in less than 5 seconds.

Provider (Joshua)

- NFR0: The providers will monitor the efficiency of their creation tools.
- NFR1: Providers responsiveness to customer reviews and engagement.
- NFR2: Providers will organize their content (watchlists, recommendations) in a clear and user-friendly manner.
- NFR3: Providers will maintain quality and accuracy of provided content.

System-Admin (Abel)

- NFR0: Should log and display load times to keep track of efficiency.
- NFR1: Admin access must require multi-factor authentication.
- NFR2: System should implement an automatic restart during a specific time every week to ensure maintenance.
- NFR3: Capable of supporting extreme number of users using at the same time without much decrease in performance.

Scenario 1: User Saves a Watchlist (Foday)

Users – Regular User (Customer)

User Saves a Watchlist

Initial Assumption: The user has access to the web app, is logged in, and is on the Watchlists page.

Normal:

- The user navigates to the Watchlists page.
- The user selects a category or name (e.g., Anime, Manga) of the watchlist.
- The user searches through watchlists.
- The user saves the watchlist.
- The system confirms the watchlist has been created successfully.

What Can Go Wrong: The user may accidentally leave the page before saving. A confirmation popup will ask if they want to exit without saving.

Other Activities: The user can edit, rate, comment, or review the watchlist later.

System State on Completion: The watchlist is now visible in the user's library and available for interaction.

Scenario 2: Provider Creates a Watchlist (Joshua)

Users – Provider

ii. Provider Creates a Watchlist

Initial Assumption: The user has access to the web app, is logged in as a Provider, and has a watchlist idea to share.

Normal:

- The user navigates to the "Create Watchlist" page.
- The user inputs a name for the watchlist and selects a category (e.g., Anime, Manga).
- The user searches for titles and adds them to the watchlist.
- The user saves the watchlist.
- The system confirms the watchlist has been created successfully.

What Can Go Wrong: The user may experience network issues, preventing watchlist submission. An auto-save feature will periodically save progress.

Other Activities: The user can modify or delete the watchlist later.

System State on Completion: The watchlist is now visible on the user's profile for others to view.

Scenario 3: Administrator Moderates a Review (Abel)

Users – System Administrator

iii. Administrator Moderates a Review

Initial Assumption: A review has been reported by multiple users for violating community guidelines.

Normal:

- The administrator logs into their admin panel.
- The administrator reviews the flagged content.
- The administrator decides whether to keep or remove the review.
- If removed, the system notifies the original reviewer.
- The action is logged for record-keeping.

What Can Go Wrong: An administrator may mistakenly remove a review. A review history log will allow them to restore it if necessary.

Other Activities: Admins can also suspend users who repeatedly violate guidelines.

System State on Completion: The flagged review is either removed or reinstated based on moderation policies.