Software Requirements Specification (SRS) Document

GitHub Repository: https://github.com/DeadeyeFo/CSC340--Group-1

05/7/2025

Version 2

Foday Mami, Abel Moran, Joshua Wusu

Table of Contents

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 the name (e.g., Anime) of the watchlist.
- The user searches through watchlists.
- The user subscribes to the watchlist.
- The system confirms the watchlist has successfully subscribed.

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.

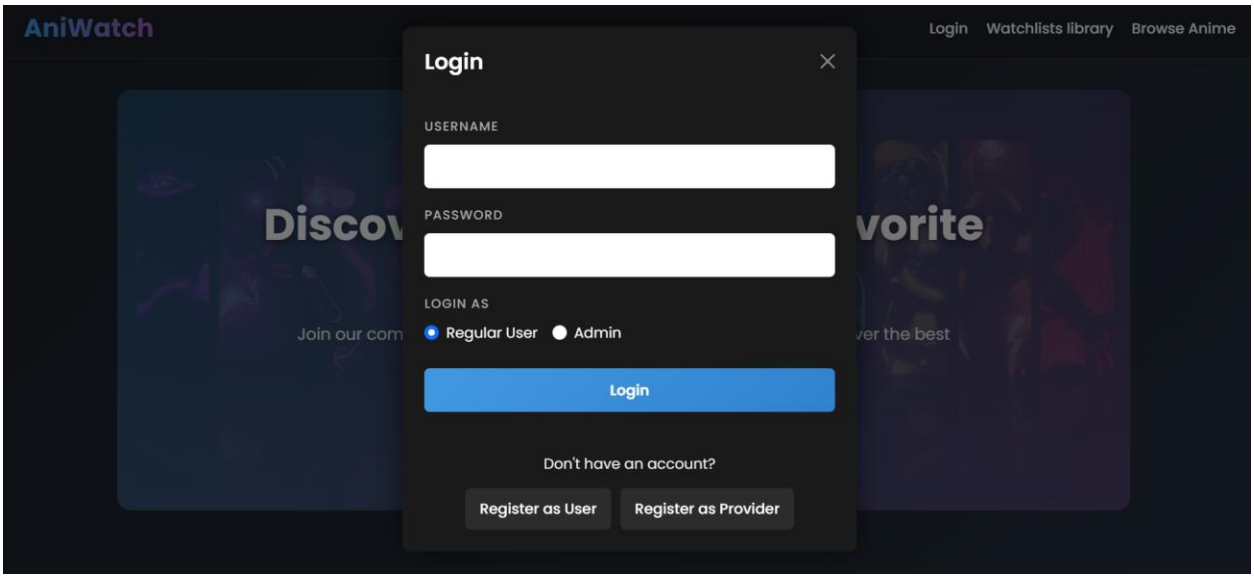
Other Activities: Admins can also suspend users who repeatedly violate guidelines.

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

Scenarios With Screenshots

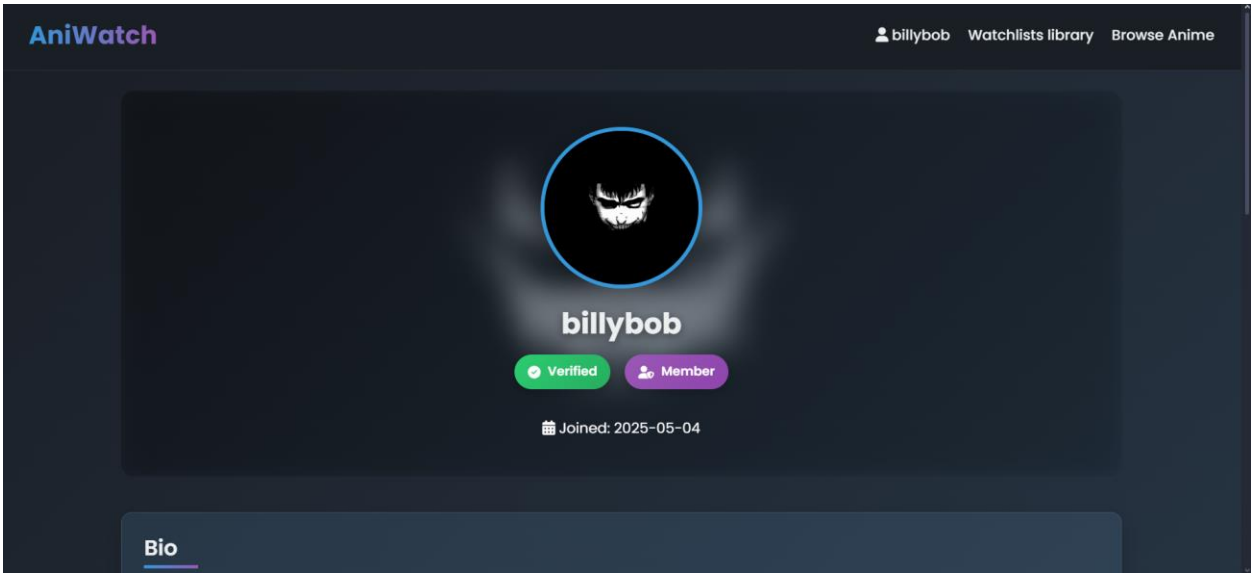
Provider: Create Profile & Create Watchlist use cases (Joshua)

1. Provider P1 opens the app, logs in for the first time. Goes to account to complete bio information.
    i. P1 fills in: name, bio, profile image, and clicks Save.
2. P1 navigates to “New Watchlist”, enters title (“Top Spring Anime”), description, selects a thumbnail, and picks seasons/titles (Anime A, B, C).
    i. P1 clicks Create and is redirected to their provider profile, seeing “Top Spring Anime” listed.
3. P1 logs out.



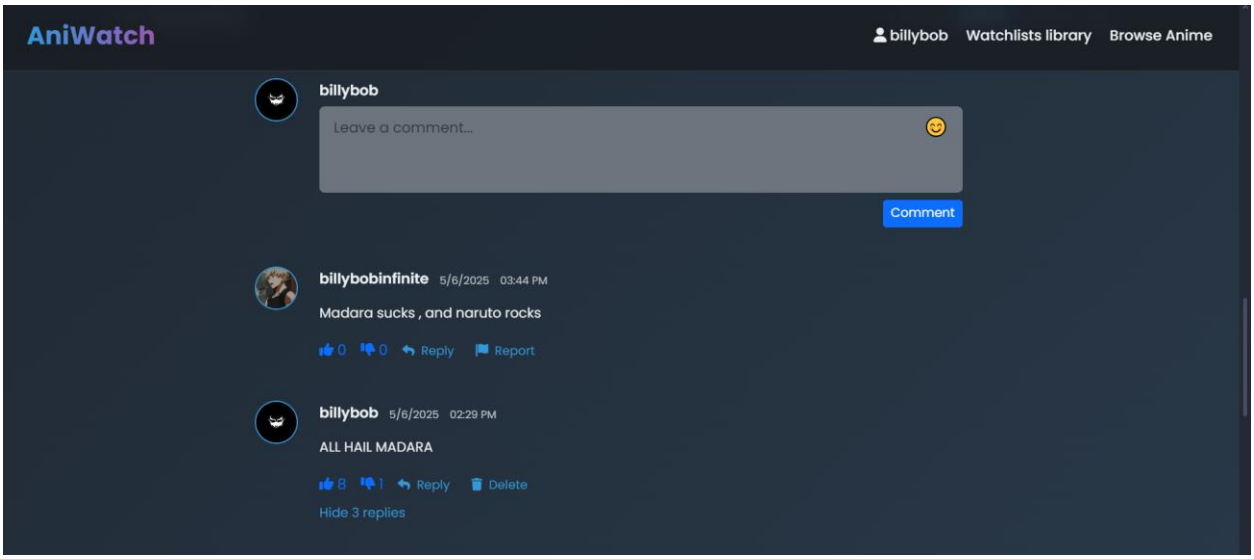
User: Create/Modify Profile & View Watchlists use cases (Foday)

1. User U1 opens the app, registers (username U1, password), and completes profile (avatar).
2. U1 browses All Watchlists and sees P1's "Top Spring Anime" with thumbnail and stats.
3. U1 clicks View on "Top Spring Anime", lands on the watchlist page showing all entries and provider info.
4. U1 edits their profile to change avatar—updates successfully and logs out.



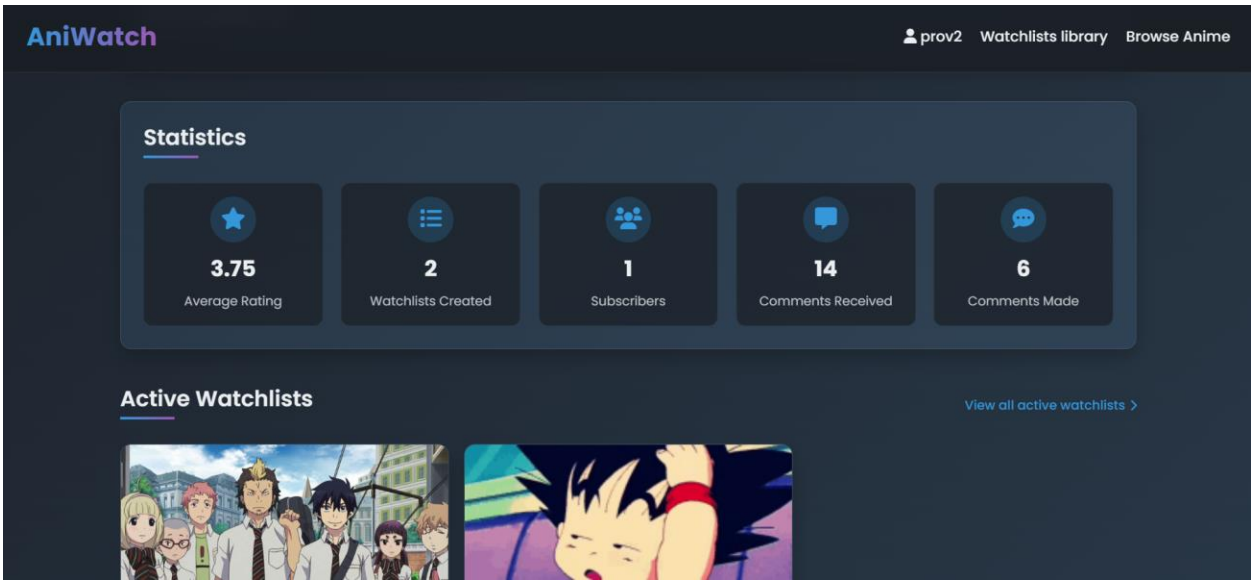
User: Subscribe, Comment & Rate use cases (Foday)

1. User U2 logs in and goes to “Top Spring Anime”.
2. U2 clicks Subscribe, sees the subscriber count increment immediately.
3. U2 writes a comment: “Love this lineup!” and submits—comment appears under the list.
4. U2 clicks on the star-rating widget and selects 5 stars—rating updates in real-time.
5. U2 logs out.



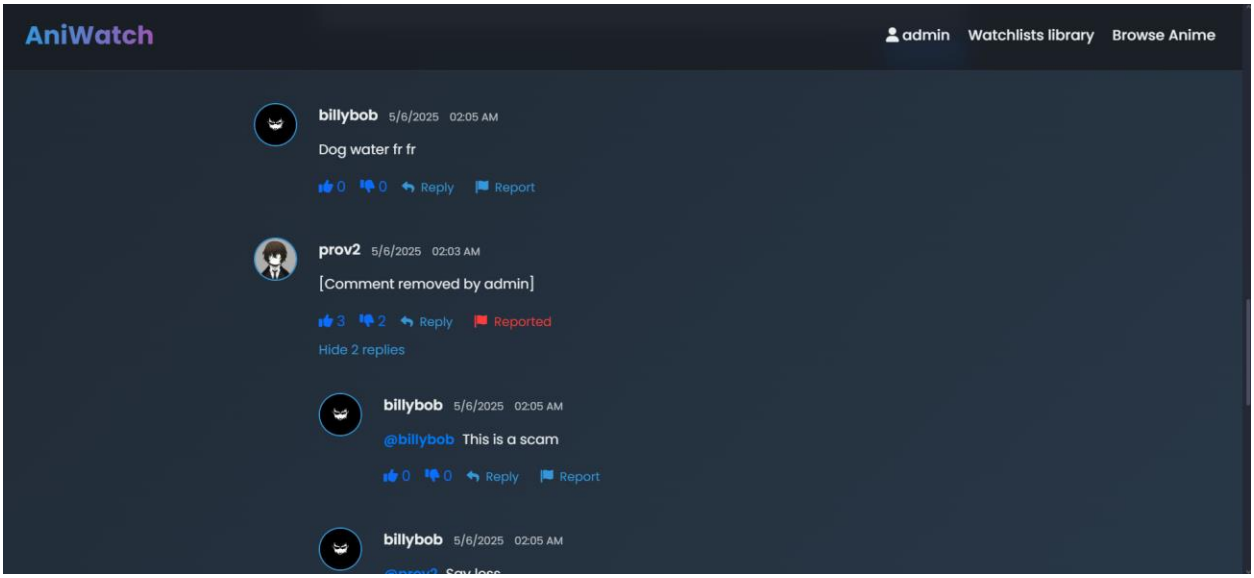
Provider: Reply to Comment, View Watchlist Stats & Modify Watchlist use cases (Joshua)

1. P1 logs back in, navigates to Profile Settings, and Views Active Watchlists. Clicks View list for specific list.
2. P1 clicks Reply on U2's comment, types "Thanks for subscribing!" and submits—reply nests under U2.
3. P1 opens the Profile Settings and views Statistics tab, sees current views, subscriber count, and average rating.
4. P1 clicks Edit, changes the description and deletes one entry, then clicks Save.
5. P1 logs out.



SysAdmin: View Usage Statistics, Manage Watchlists, Comments & Users use cases (Abel)

1. Admin A1 logs in to /admin, lands on Dashboard showing total users, providers, watchlists, comments.
2. A1 clicks Server Logs, verifies recent log lines appear (startup, login events). [Not working]
3. A1 opens Comments page, spots an inappropriate comment by U1, clicks Delete—the comment either disappears or is marked "[Comment deleted]".
4. A1 opens Users, finds U1, clicks Disable; U1 can no longer log in.
5. A1 opens Watchlists, selects "Top Spring Anime" and clicks Delete—it's removed from the list and database.
6. A1 returns to Dashboard, re-runs Dashboard stats and sees counts updated accordingly. [Not working]



Design Document

AniWatch

5/7/2025

Version 2

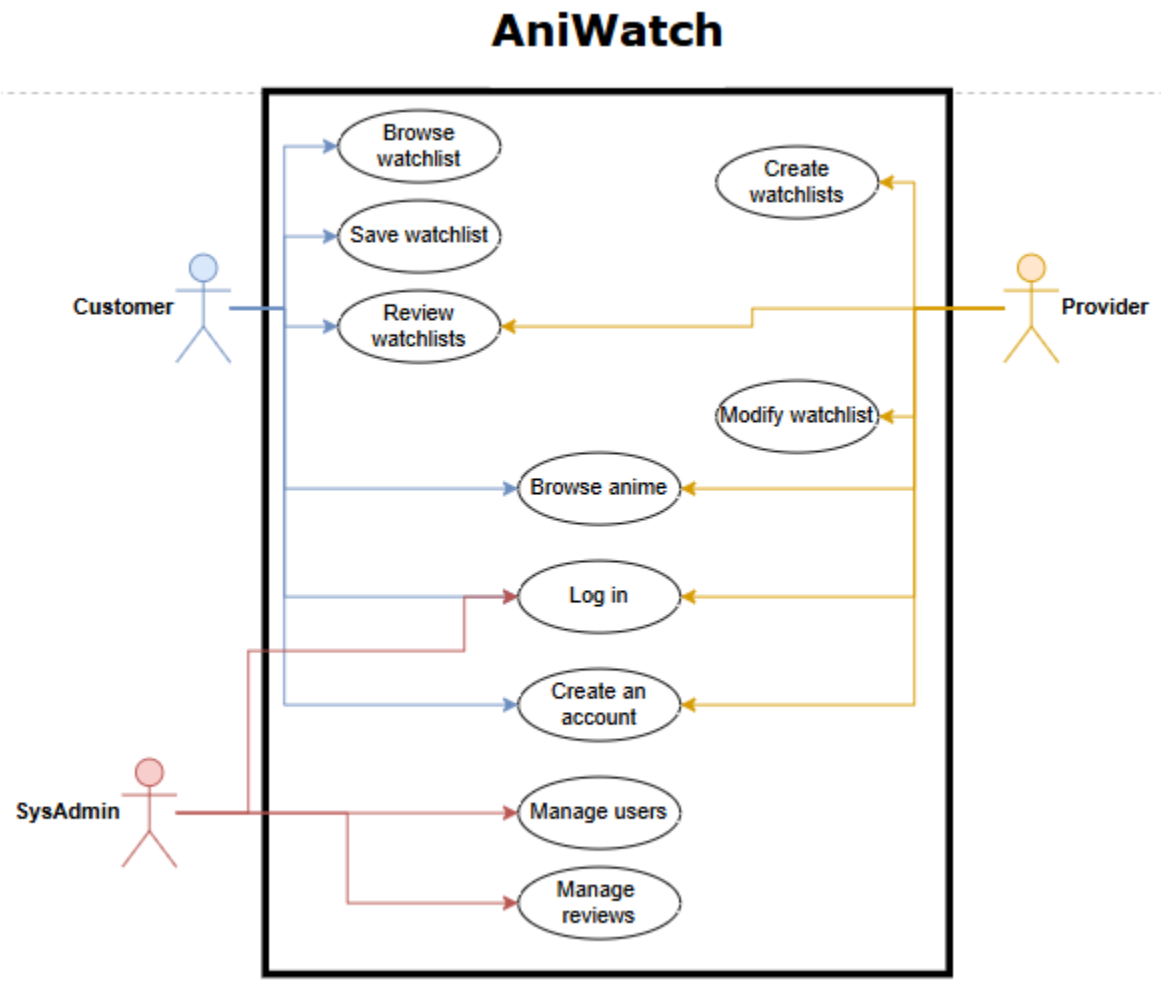
De

Isaac Foday Mami, Joshua Wusu, Abel Moran

Project Overview

The goal of the AniWatch web application is to introduce users to new anime series to watch in the form of watchlist. Watchlists can provide structure for what shows to watch and even for simplifying series that have convoluted stories where prequels and sequels are not sequential in the story. Watchlist Creators create watchlists of anime that relate to each other in some way, with the creator able to describe what the reasons for the watchlist are. Creators are able to view anime cards and add them to watchlists then display them to users. Users are presented with the watchlists and have the option to save them in their library and leave reviews on them. Users can filter through which watchlist is shown to them based on genre. Users can also add reviews for watchlists. Admins are responsible for content, review, and server moderation.

Use-Case Model



Database Schema