**Team Streams (10) milestone one**

Project: Yomiru

Simon Fraser University

CMPT 276: Introduction to software engineering

Parsa Rajabi

Submitted to Vincent Hu

Date

1 July 2025

Yadav Singh (Project Manager)

Eric Stad (JavaScript Development lead)

Jaskirat Kaur (Design/UX lead)

Karan Passi (Quality Assurance Tester)

<https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams>

**SDLC model**

The SDLC Model this project will be following is the Agile SDLC model. This model was chosen due to the iterative nature of it, as well as being more flexible breaking down the whole into smaller manageable pieces for team members to work on. The iterative cycles are focused on delivering **working** software which is the main goal of this project.

**API descriptions**

1. Myanimelist API (MAL) - (No changes from M0):

The MyAnimeList (MAL) API gives us the developers access to MAL’s extensive databases, with extensive data on Manga, anime (100 000+), and some useful features to our application. These being genre and other tags related to the content, access to existing user profiles and by extension user data, other feature that is provided is release dates for upcoming manga and anime adaptations.

MyAnimeList (MAL) is a free to use API.

Feature summary:

* Used to access latest release data and related news of anime related content. (1)
  + +This will benefit the user by informing them with info about the anime world so they can stay engaged in our app and learn more about the world of anime
* Used to access tags on existing anime and manga content in conjunction with our algorithm to suggest new anime/ manga for consumption. (2)
  + This will benefit the user by showing them new anime that they have never seen before but is similar to anime they like (via the tags and our algorithm). This will help the user even if they are an ‘anime veteran’.
* To access existing user data to help with suggestions as well as adding suggested directly onto users MAL watchlist. (3)
  + This will benefit the user by adding a seamless process to add anime to their watchlist on a reputable anime website, all automatically.

1. Trace Moe (API) (No changes from M0):

Trace Moe API is a reverse image search engine designed for taking in anime senses and finding the anime that the scene originates from. Trace Moe achieves this taking the image and identifying the anime of origin, as well as the episode number, approximate timestamp within the episode. Working in conjunction with MAL this API will also allow for us to provide a user the description of anime the image is from using the extensive databases of the MAL API.

Trace Moe API is free to use.

Feature summary:

* Given an image from an episode of an anime returns the anime of origin.
  + This will benefit the user by showing what anime an image is from, making the app more sophisticated and helpful to anime users
* Gives the exact time stamped clip from the episode allowing for the user to reexperience the clip before making a choice to take further action.
  + This will benefit the user by giving them a glimpse into the image they uploaded with a short anime clip, ultimately helping them make a choice if they want to watch the anime, and making the app more sophisticated and helpful to anime users
* Combining with MAL API to find the anime the scene is from and using the data from MAL give a summary/ synopsis of the anime this is meant to be a way for the user to get a better idea of the anime instead of wasting time watching an anime based of one clip that may have been the best scene in the show.
  + This will benefit the user by giving them a quick summary of whatever image they have. This will be very useful if they have an image of an anime clip, but they don’t know what anime it is from, and the MAL summary will give them a good idea of what this unknown anime is about

Risk Assessment

Low Risk Issues

1. UI/UX errors – misaligned buttons  
   *Mitigation:* Perform thorough UI testing on multiple devices and screen sizes; use CSS frameworks with responsive design; implement peer reviews for UI changes.
2. Minor code errors – incorrect tag closure  
   *Mitigation:* Use code linters and formatters; integrate automated code checks in the CI/CD pipeline; conduct regular code reviews.
3. Slightly slow animations – aesthetically unpleasing  
   *Mitigation:* Optimize animation code; test performance on different devices; limit animation complexity where necessary.
4. Typos in content – minor spelling mistakes  
   *Mitigation:* Proofread all text content; use automated spell-check tools; allow content review by team members or beta testers.
5. Un-factual content – content not accurate but webpage functional  
   *Mitigation:* Verify all factual information before publication; involve subject-matter experts to review content; update content regularly as needed.

Medium Risk Issues

1. UX failures – animations not functioning  
   *Mitigation:* Implement fallback static content; test animations on all supported browsers; use progressive enhancement techniques.
2. Invalid links to external sources – stale or broken links  
   *Mitigation:* Periodically run link-checking tools; have a process for timely updating or removing broken links; use redirects where possible.
3. User loss due to unclear directions for use  
   *Mitigation:* Design clear onboarding/tutorials; perform user testing to identify confusing areas; collect user feedback and iterate.
4. Third-party API dependency – only using one API for all features  
   *Mitigation:* Consider integrating backup APIs or alternative data sources; implement graceful degradation when API is unavailable; monitor API health status regularly.
5. Dependency on External APIs for Core Features  
   *Mitigation:* Design core features to be resilient to API downtime; cache API responses where possible; negotiate SLA or usage agreements with API providers

High Risk Issues

1. Exploits in third-party APIs used  
   *Mitigation:* Regularly monitor security advisories for used APIs; apply necessary security patches or updates; restrict API access with secure keys and rate limits.
2. Inappropriate user content  
   *Mitigation:* Implement content moderation filters; enable user reporting and quick removal of inappropriate content; apply community guidelines and terms of use strictly.
3. API free quota used up for MyAnimeList (MAL) and/or TraceMoe  
   *Mitigation:* Monitor API usage closely; implement caching and limit requests; consider upgrading to paid API plans or alternative services.
4. Inappropriate or Spoiler Content via TraceMoe  
   *Mitigation:* Filter content returned from API; provide spoiler warnings to users; allow users to opt out from spoiler content.
5. API Schema or Endpoint Changes (Breaking Changes)  
   *Mitigation:* Keep track of API versioning and announcements; design flexible API integration code; establish quick update and rollback procedures.

**MVC Diagram A screenshot of a diagram

AI-generated content may be incorrect.**

**Data Flow Diagram**

**A diagram of a company

AI-generated content may be incorrect.**

**Story Board**

[**https://www.figma.com/design/xZYKlFn0dYgbWf2klHPyhe/Mid-fidelity-wireframe?node-id=0-1&t=bf1zgVb4pgHmBqXI-1**](https://www.figma.com/design/xZYKlFn0dYgbWf2klHPyhe/Mid-fidelity-wireframe?node-id=0-1&t=bf1zgVb4pgHmBqXI-1)

**WBS**

|  |
| --- |
|  |
|  |
|  |  |  |
|  |  | **Schedule**   |  |  |  | | --- | --- | --- | | **🟢** | **#milestone 1 issues (13-27)**  **-4th July** | **Docs: Complete m1**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/milestone/1**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/milestone/1) | | **🟢** | **#milestone 1 issue**  **-7th July** | **M1 – video**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/22**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/22) | | **🟡** | **# Frontend v1**  **- 20th July** | **Front-end based on the wireframe v1 done**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/40**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/40) | | **🟡** | **# milestone 1.5 issue**  **-25th July** | **Milestone 1.5 checking** | | **🟡** | **# Frontend v2**  **-31st July** | **Frontend-backend sync** | |

|  |  |  |
| --- | --- | --- |
| **🟢** | **#31–33 Tracemoe feat 1–3**  **-18th July** | **Base feature integration**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/30**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/30)  [**http://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/31**](http://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/31)  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/32**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/32) |
| **🟢** | **#36 Summaries integration**  **-20th July** | **NLP-heavy and ties into core idea**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/33**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/33) |
| **🟡** | **#35 Trending board**  **-25th July** | **Builds on data from above**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/34**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/34) |
| **🟡** | **#34 Swipe feature 2**  **-27th July** | **Enhances UX**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/34**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/34) |
| **🟡** | **#37 Design interactive comps**  **-31st July** | **Frontend-backend sync work**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/37**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/37) |
| **🟠** | **#38 Unit testing v1**  **4th August** | **Validate work done so far**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/38**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/38) |
| **🟠** | **#39 Deployment**  **8th August** | **Only after tests pass and core features ready**  [**https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/39**](https://github.com/CMPT-276-SUMMER-2025/final-project-10-streams/issues/39) |
|  |  |  |

**Appendix**

**Changelog**

**Date: [July 4th]  
Updates Since Last Meeting**

1. **Wireframe Modifications**
   * **Updated the wireframe to include a fully developed search bar section.**
   * **Removed or simplified certain elements to streamline the interface.**
2. **Website Skeleton Development**
   * **Initiated the website’s structural development.**
   * **Approximately 20% of the front-end skeleton has been completed, including foundational HTML/CSS setup and partial component layout.**