

Name: Kenechukwu Maduabum

Date: February 28<sup>th</sup>, 2024

Course/Assignment: CS 3380 Project Phase 1

## **ANIME WATCHLIST MANAGEMENT SYSTEM**

- 1) Problem Statement: Anime is a popular style of Japanese animation known for its diverse genres. The anime watchlist management system allows multiple users of the system to manage and track the anime they watch. This database application is essential for anime fans seeking organized and accessible ways to manage their watchlists, find new anime to watch, share their opinions on their favorite anime, as well as create and share their recommended sub-lists to others.
- 2) ER Diagram (**There is also a standalone view in attached PDF**):
  - a) Design description:
    - USER:
      - (1) The USER entity type is one of the pillars of the database. It represents the anime fan that wants to use the application to manage their watchlist. This entity type stores many derived information about a user's anime viewing history.
      - (2) The MANAGES relationship between USER and LIST allows the user to manage and track their watchlists, as well as derive the viewing history information.
      - (3) The WRITES relationship between USER and REVIEW allows the user to share their opinions on anime through comments and ratings.
    - LIST:
      - (1) The LIST entity type is another one of the pillars of the database. It represents the watchlist of a user and stores information about the added anime of a user.
      - (2) The MANAGES relationship between LIST and USER allows a watchlist to be connected/traced to a particular user.
      - (3) The CONTAINS relationship between LIST and ANIME allows a watchlist to store multiple anime specified by a user.

- ANIME:

- (1) The ANIME entity type is another one of the pillars of the database. It stores various detailed information about a particular anime.
- (2) The CONTAINS relationship between ANIME and LIST allows an anime to be managed in multiple watchlists.
- (3) The FEATURES relationship between ANIME and REVIEW allows users to add reviews and ratings. It is also used to derive an anime's current average rating.
- (4) The HAS relationship between ANIME and EPISODE allows the application to connect an anime to its many episodes and derive its number of episodes.
- (5) The INCORPORATES relationship between ANIME and CHARACTER allows users to see stored detailed information on the characters that are featured in an anime.

- REVIEW:

- (1) The REVIEW **weak entity type** represents a user's opinion on an anime. It stores a user's numerical rating and comments for a particular anime. A user's review is not unique, but in the context of a specific anime it is.
- (2) The WRITES relationship between REVIEW and USER allows reviews for an anime to be traced back to the creating user.
- (3) The FEATURES **identifying relationship** between REVIEW and ANIME allows a user's review entry to be traced back to an anime, in order to provide uniqueness to the review.

- EPISODE:

- (1) EPISODE is a minor entity type that stores basic information about an anime episode.
- (2) The HAS relationship between EPISODE and ANIME allows an episode's information to be related to its anime.

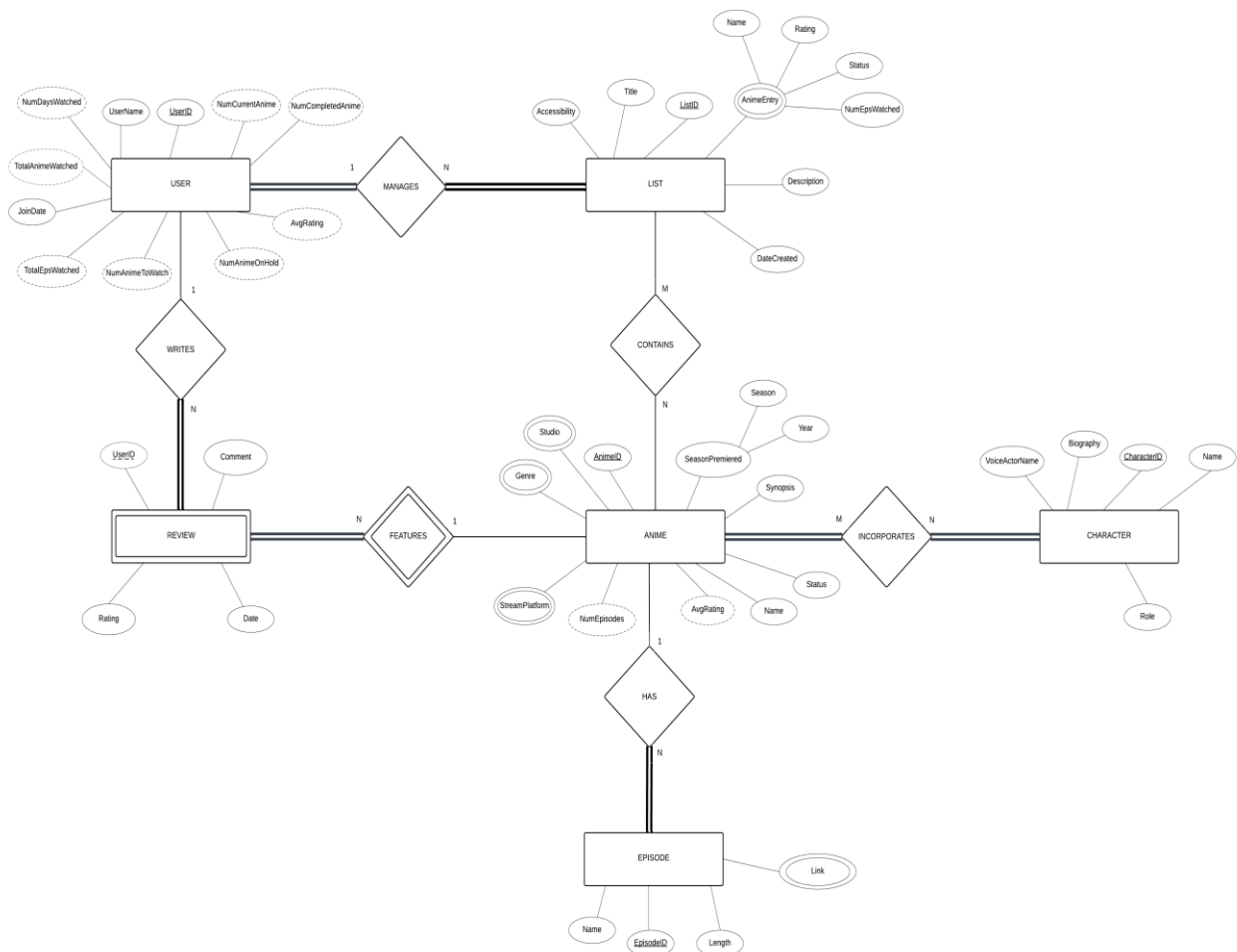
- CHARACTER:

- (1) CHARACTER is a minor entity type that stores basic information about a featured character in an anime.

(2) The INCORPORATES relationship between CHARACTER and ANIME allows a character's to be connected to the anime they feature in.

b) Assumptions Made:

- A user can write multiple reviews, but only for a single anime.
- All users will have at least one list which will be set to private by default and can be changed to public if the user decides. They could have more through creating sub-lists that are public by default (Total participation / 1:N).
- A character can be part of multiple anime due to things like collaborations or sequels (M:N).
- An anime could be part of 0 lists (partial participation) to many lists (1:N)
- Lists could be empty if it was created, but a user didn't add any anime to it (partial participation).
- Not all anime could feature reviews or episodes yet (partial participation).
- Anime names only have 1 translated English name (not multivalued).
- Anime with sequels are not related to each other and are assumed to be separate anime.
- The watch-activity statistics of a user are always public, and automatically derived when visiting a user's profile.



### 3. Functions provided:

#### a) Description: **Anime search filtering**

- Entities involved: USER, ANIME, REVIEW.
- Roles and details: USER can filter searches for ANIME. ANIME stores information on its Genres, Status of completion, Studio, and SeasonPremiered which are all useful for carrying out diverse searches. The AvgRating of an ANIME can also be used as a filter and is derived by taking the average of all the ratings from REVIEW.

#### b) Description: **Anime watchlist management**

- Entities involved: USER, ANIME, LIST, REVIEW
- Roles and details: A USER can manage the ANIME in their LIST. A USER can view the page (information) of an ANIME and add it to their LIST by either using a toggle when

viewing an ANIME, or automatically by making a REVIEW. A LIST is used to store information on each ANIME that was added by a USER. The information includes: the number of episodes watched, watch status, and the personal rating of each ANIME they added. A USER can also remove an ANIME from their LIST.

c) Description: **Anime rating and review**

- Entities involved: USER, REVIEW, ANIME, LIST
- Roles and details: A USER can REVIEW an ANIME in their LIST and ANIME stores/features multiple REVIEWS from different USERS. REVIEW stores the Rating and Comment that a USER put out for a specific ANIME. A USER needs to add an ANIME to their LIST in order to write a REVIEW, if not, then the ANIME is automatically added to the USER's default LIST.

d) Description: **Creation of recommended sub-lists**

- Entities involved: USER, LIST, ANIME
- Roles and details: A USER can create additional sub-LISTS (which must be a subset of the ANIME in their default LIST) based on their own personal criteria and preferences. The default LIST of a USER will be private by default, and additional LISTS that USERS create will be set to public by default and featured on their profile.

e) Description: **Running reports on personal anime viewing history**

- Entities involved: USER, ANIME, LIST
- Roles and details: A USER profile automatically derives and shows stats of their ANIME watching activity from their LIST. USER profile stores NumWatchedDays, TotalAnimeWatched, etc. which are all derived from the ANIME in their default LIST when anyone accesses their profile.