Project Name: Movie Recommendation System **Team#** 9

Test Stage: **Unit**

Test Date: **12-02-2025**

Test Case ID#: **RF_T3**

Name(s) of Testers: David

Test Description: Testing file writing and rating functionality with malformed but accessible my_ratings.txt, and Reviews for rated movies and unrated movies. Testing updating lines in my_ratings.txt and adding lines in a malformed my_ratings.txt. Testing to see if malformed lines get ignored when loading reviews into system. (Will create a review with a movieID that matches a malformed line). Also tests acceptance criteria (Test with invalid movieID and invalid rating).

Indicate where are you storing the tests (what file) and the name of the method/functions being used.
Test code is within testing directory inside RatingFunctionalityTest.java. testLoadMovies(), createReviewObject(), and testUpdateReviews() are being used.

Automated: ___ **no** ___

Results: **Pass**

Preconditions for Test: Files must exist and be accessible in /data directory in CSV format.

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Load movies into system	movies.txt	Movies loaded into system	Movies loaded into system	Done since creating reviews is dependent on MovieDB
2	Create Reviews and store in ArrayList	my_ratingsfwt3.txt	Review objects are created and stored inside ArrayList	Review objects are created and stored inside ArrayList	Created a mix of new reviews and ones to update. Also created a review with a movieID matching a malformed line to test if it is treated as new.

					And created Reviews with invalid movieID’s and invalid rating.
3	Call testUpdateReview(reviews)	fw_t3 ArrayList of reviews	2 lines of my_ratings.txt are updated, while 2 lines are added and Malformed line, invalid ratings are ignored.	Appropriate lines are updated, and 2 lines are added. Malformed line, invalid ratings are also ignored.	

Post condition(s) for Test: my_ratings.txt reflects updated and new line changes appropriately.