

Project Name: Recommendation System**Team# 9****Test Stage:** System**Test Date:** 02/12/2025**Test Case ID#:** FR-RE-T5**Name(s) of Testers:** Marko**Test Description:**

Testing the recommendation feature. Testing if recommendations are not the same as the user rated movies.

Indicate where are you storing the tests (what file) and the name of the method/functions being used.

Test resides in recommendationTest.java file under the testing directory in milestone2. loadMovies(), add(), makeReviews(), ratingsFileExists(),readRatingsFile(),getMovie(),recommendMovies() are being used.

Automated: no**Results:** Pass

Preconditions for Test: Empty ratings file, Ratings file is formatted correctly

Step #	Test Step Description	Test Data	Expected Result	Actual Result	Notes
1	Add reviews	List of rated movies: "Forrest Gump"	All movies are in my_ratings.txt and have a 5 star rating	All movies were in my_ratings.txt and had a 5 star rating	Make sure your my_review.txt is empty of reviews before running test.
2	Initialize User object	Reviews list containing 1 reviewed movie and moviesInReviews list 1 movie.	User object with reviews list containing the reviewed movies and the reviewedMovies list containing the movies.	User object with reviews list containing the user's review and the reviewedMovies list containing the movie "Forrest Gump".	

3	Initialize RecommendationEngine object	User object	Recommendation Engine object with user instance variable.	Recommendation Engine object with user instance variable.	
2	Call recommendMovies()	my_ratings.txt, reviewedMovies arraylist	Movies that do not have title “Forrest Gump”.	“Avengers: Endgame”, “The Dark Knight”, “Avengers: Infinity War”, “Inception” and “Interstellar”.	

Post condition(s) for Test:

my_ratings.txt has the new ratings. Reviewed movies are stored in the reviewedMovies arraylist in User. Recommendations are printed to console.