**Test Plan: Movie Collection**

**Functional Testing:**

* Verify the app renders correctly in latest Chrome browser on Windows, MacOS, Android, and iOS (Selenium Webdriver or Appnium)
* Verify a registered user is able to enter new movies into the collection.
* Verify user is able to search existing movies by these fields: Genre, Actors, Title, Year, Rating
* Verify user is able to see a list of all movies in user's collection.
* Verify user is able to delete a movie.
* Verify user is able to update a movie.
* Verify user is able to record the following info about any given movie: Genre, Actors, Title, Year, Rating

**Database Testing:**

Verify data integrity when the same movie is posted more than once (there should not be duplicate records).

**Performance Testing:**

* Measure the performance of register, login, movies, and users API routes.
* Verify performance expectations are met (from Product Owner) when X number of users access the app/server (jMeter Script)

**Security Testing:**

* Verify that user tokens are different for different users.
* Verify that user tokens are different for each request.
* Verify that a user cannot modify another user’s movie.
* Verify that a user cannot delete another user’s movie.

**Risks and Assumptions:**

* Current user story does not document behavior of user registration/logon messages.
* Current user story does not document performance requirements.
* Current user story does not document security requirements.

**Backend Unit test coded in server/test/test.js:**

**User Registration:**

* It should register a new user.
* It should fail to register with an email already taken.

**Login:**

* It should login successfully.
* It should send back an unauthorized error.
* It should send back a user not found error.

**Movie Tests:**

Create Movie

* It should create a new movie with the API.

Get Movies

* It should send back a list of all movies.
* It should send back a list of all movies created by a specific User.
* It should send back a list of all movies queried by genre.
* It should send back a list of all movies queried by actors.
* It should send back a list of all movies queried by title.
* It should send back a list of all movies queried by year.
* It should send back a list of all movies queried by rating.

Update Movie

* It should update a movie given a valid movie id.
* It should NOT update any movie with an invalid movie id.

Delete Movie

* It should delete a movie given a valid movie id.
* It Should NOT delete a movie with an invalid movie id.

**Frontend testing coded in Selenium/Mocha/Node.js:**

**User Registration:**

* It should register a new user.
* It should fail to register with an email already taken.

**Login:**

* It should login successfully.
* It should send back an unauthorized error.
* It should send back a user not found error.

**Movie Tests:**

Create Movie

* It should create a new movie with the API.

Get Movies

* It should send back a list of all movies.
* It should send back a list of all movies created by a specific User.
* It should send back a list of all movies queried by genre.
* It should send back a list of all movies queried by actors.
* It should send back a list of all movies queried by title.
* It should send back a list of all movies queried by year.
* It should send back a list of all movies queried by rating.

Update Movie

* It should update a movie given a valid movie id.
* It should NOT update any movie with an invalid movie id.

Delete Movie

* It should delete a movie given a valid movie id.
* It Should NOT delete a movie with an invalid movie id.

**Backend Apache jMeter v3.3 Tests**

These will be useful for determining performance metrics going forward.

**User Registration:**

* It should register a new user.

**Login:**

* It should login successfully.

**Movie Tests:**

Create Movie

* It should create a new movie with the API.

Get Movies

* It should send back a list of all movies.
* It should send back a list of all movies created by a specific User.
* It should send back a list of all movies queried by genre.
* It should send back a list of all movies queried by actors.
* It should send back a list of all movies queried by title.
* It should send back a list of all movies queried by year.
* It should send back a list of all movies queried by rating.

Not currently coded as route has not been defined in source code.

Update Movie

* It should update a movie given a valid movie id.

Delete Movie

* It should delete a movie given a valid movie id.