

Iteration 1

Team Name: GUO

Ahmed Alzubairi (afa2135), Michael Stone (ms5512), Mauricio Antonio Quintanilla (maq2119), Christian Kowalczyk (cjk2159)

Part 1. Team github repo

<https://github.com/AhmedAlzubairi1/GUO>

Part 2. User stories and Acceptance testing (20 points max)

12 points max for there should be acceptance tests for each user story (the user stories given in this report, not necessarily the same as the revised proposal). There should be at least one test with valid input and at least one test with invalid input. If a user story could not possibly involve invalid inputs, that should be explained. Take off 2 points for each user story without a valid test case and two points for each user story without an invalid test case or explanation, up to 12 points off (i.e., 0 for this portion of the 20 points total).

This is to account for 3 to 5 user stories, e.g., if there are 5 user stories and none of them have any acceptance test cases, this part would be 0/12.

8 points max for the discussion of the acceptance testing process and bugs found/fixed. 0/8 if this discussion is missing, prorated partial credit among the user stories covered or not covered by this discussion.

User Stories:

<Creating an account> As a movie hobbyist, I want to be able to create an account with my Gmail for a movie related website of my interest.

<Organizing List> As a movie hobbyist, I want a place to store movies on my watch list so that I can have an organized list of my favorite movies. My condition of satisfaction is that I am able to search for any movie and be able to have it into my list.

<Anticipating Top Movies> As a film critic, I want to know what the top movies are. My conditions of satisfaction is that if I click some button or some functionality, I should be able to see some of the top few IMDb movies.

<Movie Information> As a busy person I want to know if a movie has a good plot before watching it so that I don't feel like I wasted my time watching a bad movie. My condition of satisfaction is that I can search for a movie on the website and be given some rating, plot, or other information.

<Meeting for movie> As a solo movie watcher, I want to be linked with others who plan to watch a movie, so that I can watch it virtually with them. My condition of satisfaction is that I am able to get a message or some sort of notification for a Zoom Link where I can watch the movie with others.

Acceptance Testing & Discussion of Acceptance Testing:

<Creating an account> User goes to the home page at the local host (<https://127.0.0.1:5000/>) and clicks the Google Login in navbar.

- a. In a valid input, a user enters a valid Gmail and password, creating an account and logging them in. Users are signaled that they are logged in because they get redirected to the home page with their profile picture and name in the Navbar.
- b. In an invalid input, a user inputs an invalid Gmail and password. A MovieLink account is not created, and the user is alerted by Google that the login is not correct. They are not taken to the user homepage.

During our testing:

Valid Input: User went to the home page and logged in with valid credentials.

We inputted ahmedalzubairi1@gmail.com and password=correct_pass into the Google login screen. We were taken to the user home page and saw the name Ahmed Alzubairi displayed in the Navbar. We saw a blank list of movies & a blank list of friends as it is a new account. There were **no bugs** to report when we tested this.

Invalid Input: User went to the home page and clicked login, but had incorrect password.

We inputted ahmedalzubairi1@gmail.com but with the password=incorrect_pass to the login screen. We were alerted by Google login form that the login was invalid. There were **no bugs** to report when we tested this.

<Organizing List> User logs in, and is taken to the user home page with a blank wish list of movies. They want to add to this list.

- a. In a valid input, they can use the search bar to look up a specific movie, click the right movie, and click a button to add it to their wish list.
- b. In an invalid input, the user searches for a movie that doesn't exist and is taken to a screen with no movies to choose from. They can go back to the user home page or attempt to search again.

During our testing:

Valid Input: User went to the user homepage after logging in and searched up a valid movie and added to their wish list.

We searched for the movie "Shrek," and saw a list of Shrek movies. We were able to click on the original Shrek, add it to our wish list, which made it appear back on the user home screen under wish list. There was initially a bug during this acceptance testing where we attempted to display more movies than existed in the database after a given search (leading to an error in the terminal), but we were able to **fix** this bug.

Invalid Input: User went to the user home page after logging in, and searched a movie that did not exist.

We searched for a movie “asd!f--jkl”, saw a blank list because that movie does not exist. We were able to exit back to the user home page with the navigation bar. One bug to report here is when you search for a (no entry in the text bar) an error occurs. Other than that, there were no bugs on invalid inputs.

<Anticipating Top Movies> User logs in, and is taken to the user home page and sees a random selection of the top 250 movies from IMDb. They can also generate a new random set of movies.

- a. In a valid input, the user can click on a “Generate Top Movies” link below the list of top movies and it will refresh this list.
- b. There is no invalid input for this user story, because the list is auto generated on the loading of the user home page, and the “Generate Top Movies” simply refreshes this page.

During our testing:

Valid Input: We went to the user homepage after logging in and were able to see a subsection of the top 250 IMDb movies.

We clicked on the “Generate Top Movies” button, which refreshed the page and updated the list with a new set of quality movies.

Invalid Input: N/A. You can’t have an invalid input because the button is guaranteed to give you movies.

<Movie Information> User logs in and wants to see information on a valid movie that exists.

- a. In a valid input, the user can search for a movie that does exist. They will get a selection of movies with a match name, and be able to click on one to get more information.
- b. On an invalid input, the user might (again) search for a movie that does not exist, or search for a movie that has yet to have any information.

During our testing:

Valid Input: User logs in and looks for information on a valid movie that exists.

We searched for American Psycho, were given a list of movies with similar titles, clicked on American Psycho and were able to read the plot summary. There were **no bugs** to report.

Invalid Input: User looking for information on a invalid movie that doesn’t exist or doesn’t exist yet. We searched for The Matrix 4, and were returned a movie that we could click on that has yet to come out. The **bug** was that it had no information and should probably not be included in the results. We were unable to fix this for this iteration, because many movies in the IMDb database are likely similar and upcoming.

<Meeting for a movie> User wants to create a session with all of his friends to watch a specific movie and notify them in some way.

- a. In a valid input, the user can search for a movie that does exist. They will get a selection of movies with a match name, and be able to click on one and then click ‘add session’. This will send an email out to all of their friends.
- b. On an invalid input, the user might (again) search for a movie that does not exist, or search for a movie that has yet to have any information. However, in terms of adding the session itself because it is a button click there is likely no invalid input. Another possible “invalid” input is if

they were to enter a faulty link or not include the date/time. Our current system just sends the information that is given in the form though, without demanding for valid inputs.

During our testing:

Valid Input: User wants to create a session with all of his friends to watch a specific movie that exists and notify them in some way.

We searched for Psycho, were given a list of movies with similar titles, clicked on Psycho and were able to click “add session”. This sent an email out to everyone on the friends list (which was another account that we had logged into the system from the local database). There were no bugs to report.

Invalid Input: User looking for information on a invalid movie that doesn’t exist or doesn’t exist yet. We searched for The Matrix 4, and were returned a movie that has yet to come out. The **bug** was that it still had the ability to add a session, even though you would never actually be able to watch that movie with a friend.

Part 3. Automated build and unit testing (10 points max)

Tests are located:

https://github.com/AhmedAlzubairi1/GUO/tree/main/my_tests

build/package manger:

See readme

<https://github.com/AhmedAlzubairi1/GUO/blob/main/README.md>

Our project doesn’t require a build in the traditional sense because we are not compiling. We use pip as our package manager and added a requirement.txt for the installs needed. See readme for instructions on how to use. For the reports on bugs and styles, we use flake8/flake8-bugbear and made it into a sh script to automate the process.

Part 4. Style checker and bug finder (10 points max)

Link to automated bug report tool: (tool used is flake8-bugbear)

https://github.com/AhmedAlzubairi1/GUO/blob/main/create_bug_report.sh

Link to unclean bug report (we didnt have any bugs that static analysis):

https://github.com/AhmedAlzubairi1/GUO/tree/main/bug_finder_report

Link to clean bug reports:

https://github.com/AhmedAlzubairi1/GUO/tree/main/cleaned_bug_checker_report

Link to automated style check report tool: (flake8)

https://github.com/AhmedAlzubairi1/GUO/blob/main/create_style_checker_report.sh

Link to unclean check style report:

https://github.com/AhmedAlzubairi1/GUO/tree/main/style_checker_report

Link to clean check style report:

https://github.com/AhmedAlzubairi1/GUO/tree/main/cleaned_style_checker_report

