COM1028 Software Engineering

Coursework Assignment

FlickFinder

URN	6845721
Username	gb01005
GitLab Link	https://gitlab.surrey.ac.uk/csee/com1028/2023-24/com1028gb01005
Date	/05/2024

1 Personas, Scenarios and User Stories

1.1 Personas

Jane, a Film Studies student

Jane, age 20, is a Film Studies student at King's College London. Having studied film studies at both GCSE and A Level, she has developed a deep passion for films and has a keen eye for picking out good films. To pursue her passion for films further, she has started a Film Studies course at university. During her spare time, she also maintains a blog and YouTube channel where she analyses and critiques movies, as well as uploading short films that she has made. Recently, she has been hired to be a social media content creator and graphic designer for the university.

Jane's experience maintaining a blog and creating YouTube videos and short films has exposed her to many different types of software and means that she is a confident 'power user' of digital technology. Jane believes that films share a strong relationship between the societies they portray and the real-world societies during their time of release and are important documents of our time.

As part of this belief and her desire to excel in her Film Studies degree, she wants to become an expert user of FlickFinder, and use it as a platform for gaining knowledge about movie trends over time, by seeing what movies were the highest-ranking films of different years, and analyse these movies to understand why they were successful and understand how the trend of the highest-ranked movies of a year represented the society of that time.

Samantha, an ER Nurse

Samantha, age 35, is an ER Nurse at a large hospital in San Fransisco, California. She has a degree in Nursing and moved to California over 5 years ago. She is married and has 2 children. Her job requires her to work very long and stressful shifts. This coupled with having children means that she has limited free time. She is a frequent social media user, using it to keep in contact with family and friends, and so is a competent digital user. Through social media, she has also recently become a fan of Leonardo DiCaprio.

Samantha believes that she can handle most aspects of digital technology, though she does acknowledge that there are situations in which she has to seek help. Due to a lack of free time to properly search for movies, Samantha wants to utilise FlickFinder to be able to keep track of the movies she has watched and wants to watch. She wants to utilise its ability to list movies by ranking, and its ability to list the stars in a movie, to figure out which movies she may want to watch quickly. Also, as a recent fan of Leonardo DiCaprio, she hopes to use FilckFinder's ability to list the movies acted by a star, to add movies DiCaprio has acted in to the list of movies she wants to watch.

1.2 Scenarios

Preparing for Movie Night

Samantha is a busy professional with two children. She attends fortnightly movie nights with her friends, to take a break from her busy life.

As part of the preparation for movie nights, Samantha and her friends take turns organising it, and it is Samantha's turn to organise the upcoming movie night. She uses FlickFinder's ability to list movies by ranking and uses the filter to easily and quickly select movies that are not in her watched list and are genres that she and her friends like. However, Samantha also needs to be able to share a list of movies she's selected and be able to set a speculative date and time. She needs to easily add and remove movies from the list she's shared and dynamically update the chosen date and time, all in a way that's visible to her friends, due to the dynamic nature of her and her friends' work pattern.

Samantha asks on the FlickFinder subreddit to see if anyone can recommend an easy way of doing this, rather than keeping a separate Google document for the list, and constantly messaging date and time updates to her friends. She gets many replies, and the vast majority of the replies, including the most upvoted ones, all suggest that she uses FlickFinder's built-in WatchParty feature to share a movie list, so that she can dynamically change the movie list, all whilst her friends have access to the most updated version of the list and are still able to make use FlickFinder's features to learn more about the film (such as the stars in the movie, the director, etc). They also suggest that she use Google Calendar's ability to share and view other people's calendars, and to link her Google account to FlickFinder so that she can easily set a date and time and link a movie list to that time such that it is visible to all her friends, all from within her and her friends' FlickFinder app.

As FilckFinder is not linked to her Google account, she quickly links the app to her Google account, via the in-app sign-in feature, and gets her friends to share their calendars with her. She uses the WatchParty feature to select a speculative date and time and selects a movie list to be associated with that time, so that, when her friends check their Google calendar, they can immediately see the movie night, and use the link with the calendar event to see the movie list and be able to vote on which of the movies they want to watch the most.

1.3 User Stories

1. As a user, I want to search for movies by actor so that I can find the movies that my favourite actors have acted in.
2. As a film club member, I need to be able to create and share movie lists, so that I can suggest movies for group screenings and share recommendations with fellow members.
3. As a user, I want to search for movies by year so that I can find classic/older movies.
4. As a user, I need to be able to filter movies by age rating so that I can find appropriate movies suitable for my family movie night.
5. As a user, I want to be able to search for movies by directors so that I can find the movies directed by my favourite director.
6. As a user, I need to be able to sort movies by ranking so that I can quickly find a good movie to watch.

2 Critical Analysis and Reflection

2.1 Completed Requirements and Reflection

Must-have	
M.1: Retrieve a list of all the movies in the database	completed
M.2: Retrieve a movie by its ID.	completed
M.3: Retrieve a list of all the people in the database	completed
M.4 Retrieve a person by their ID.	completed

Should-have	
S.1: Retrieve a list of stars by a specific movie.	completed
S.2: Retrieve a list of movies by a specific star.	completed
Could-Have	
C.1: Add the parameter 'limit' to `list movies` and `list person route, ratings` route.	completed
C2: Retrieve a list of movies ordered by average rating in descending order	completed

The implementation of functionality, testing, and validation was successful overall. Initially, I overlooked the possibility of users entering invalid parameters but addressed this by implementing checks at both the Controller and DAO levels. I checked syntactic validity at the Controller level to prevent passing syntactically incorrect parameters, leading to an appropriate webpage and 400 status code if necessary. For IDs, I restricted the passing of integers less than 1 to the DAO, as there are no ID values less than 1 in the database.

Similarly, for the limit and votes parameters, after syntax checking, I handled edge cases and input validity within their respective DAOs. I decided not to report 400 status codes for queries, opting instead to return results as if parameters were set to defaults, providing a more seamless user experience. IMDb's advanced search filter guided this, as similarly, it doesn't allow syntactically invalid inputs to be inputted, and when presented with edge cases and invalid inputs (which can be forced by altering the parameter in the URL) it defaults to their default values (e.g. 0 for votes) (IMDb, 2024a).

Additionally, I enabled users to specify both a limit and minimum votes for ratings, recognising that users may want to retrieve a specific number of movies with more than a certain number of votes. Once again, IMDb's search filter guided this decision, whilst the search filter did not allow to limit movies, it did allow to combine filters (e.g. minimum rating and minimum number of votes) (IMDb, 2024a). Lastly, I incorporated appropriate 404 status codes to signify when no matching/valid records are found in the database, similar to IMDb (IMDb, 2024a). As for future improvements, consideration could be given to allowing users to view movies by specific directors.

2.2 Professional Aspects

This project makes use of the MVC architecture(Sommerville, 2015, p.176). This separation of the architecture, into distinct components with specific responsibilities, promotes low coupling and high cohesion. This high cohesion helps reduce program maintenance costs (Appleton, 2024c) as the time needed to understand the code is less than if it were to have low cohesion. The low coupling offered by the MVC architecture allows the API's features to be extended easily and, lets the API be used in different parts of FlickFinder without much change to the code.

Related to code maintainability, the MVC architecture and the use of DAOs encourage coding to the Single Responsibility Principle (Appleton, 2024b), making it easier to maintain consistent and meaningful variable and method names. This improves the overall readability and consistency, leading to a more professional coding format (Appleton, 2024d). Promoting the development of a more professional coding format, allows the code to be maintainable.

From initial development, there are numerous laws and licensing agreements that the API and the eventual FlickFinder application must follow (gov.uk, 2023; Appleton, 2024d),e.g. WCAG. Given the API is using the IMDb database and that FlickFinder is to be released as a public application, this means it cannot make free use of the IMDb database (IMDb, 2024b), it must contact IMDb to negotiate an appropriate license. If Flick-Finder is to be released as a non-commercial app, there are alternatives which allow free access for non-commercial use, such as TMDB (TMDB, 2024b). However, similar to IMDb, should FlickFinder decide to become a commercial app after release, it must negotiate the appropriate license (TMDB, 2024a).

Given the plethora of similar apps and their included features (Cunningham, 2021; Trickyguy37, 2022; bokunobokuu, 2023; kenziekray, 2023), it is most probable that FlickFinder will build upon the API in a manner that necessitates storing user data (e.g. movie recommendations, watchlists, etc). Doing so would require FlickFinder to comply with relevant laws and obtain the appropriate licenses (Appleton, 2024d; theInformation Commisioner's Office, 2024),e.g. the Data Protection Act (theInformation Commisioner's Office, 2018).

FlickFinder's sustainability can be assessed through its hardware and software (Appleton, Newman and Winks, 2024). FlickFinder will be comprised of multiple databases and will adopt a distributed architecture style (Appleton, 2024a). Within this style, the dynamic nature of the volume of user requests requires FlickFinder to be a serverless architecture (Appleton, 2024a). Using a serverless architecture allows for more sustainable use of hardware, FlickFinder can choose to run its services with green cloud providers (Appleton, Newman and Winks, 2024). On a software level, assuring FlickFinder is developed with sustainability in mind helps ensure the code is efficient and, minimises unnecessary interaction with the servers, contributing to less reduced usage of server hardware. This can achieved by keeping a local cache of recently viewed films, ensuring as much of the processing can be done on the device, etc (Appleton, Newman and Winks, 2024).

3 References

Appleton, J. (2024a) 'Week 4: Architecture Design. COM1028: Software Engineering.' Available at: https://surreylearn.surrey.ac.uk/d2l/le/lessons/252849/topics/2980498 (Accessed: 8 May 2024).

Appleton, J. (2024b) 'Week 5: Software Design and Implementation. COM1028: Software Engineering.' Available at: https://surreylearn.surrey.ac.uk/d2l/le/lessons/252849/topics/2980498 (Accessed: 8 May 2024).

Appleton, J. (2024c) 'Week 7: Software Evolution. COM1028: Software Engineering.' Available at: https://surreylearn.surrey.ac.uk/d2l/le/lessons/252849/topics/2986999 (Accessed: 8 May 2024).

Appleton, J. (2024d) 'Week 11: Academic and Professional Skills. COM1028: Software Engineering.' Available at: https://surreylearn.surrey.ac.uk/d2l/le/lessons/252849/topics/3005259 (Accessed: 8 May 2024).

Appleton, J., Newman, A. and Winks, O. (2024) 'Week 10: Sustainability. COM1028: Software Engineering.' Available at: https://surreylearn.surrey.ac.uk/d2l/le/lessons/252849/topics/3002275 (Accessed: 8 May 2024).

bokunobokuu (2023) ‘Is there an app where I can look for movies\shows and add them to a collection? And discover more from my favorite genres?’, *r/androidapps*. Available at: www.reddit.com/r/androidapps/comments/18gs3u5/is_there_an_app_where_i_can_look_for_moviesshows/ (Accessed: 10 May 2024).

theInformation Commisioner’s Office, (ICO) (2018) *Data Protection Act 2018*. King’s Printer of Acts of Parliament. Available at: https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted (Accessed: 8 May 2024).

theInformation Commisioner’s Office, (ICO) (2024) *Your beginner’s guide to data protection*. ICO. Available at: https://ico.org.uk/for-organisations/advice-for-small-organisations/your-beginner-s-guide-to-data-protection/ (Accessed: 9 May 2024).

Cunningham, J. (2021) *7 Apps Every Movie Lover Needs*, *Lifehacker*. Available at: https://lifehacker.com/7-apps-every-movie-lover-needs-1847352175 (Accessed: 10 May 2024).

gov.uk (2023) *Making your service accessible: an introduction - Service Manual - GOV.UK*. Available at: https://www.gov.uk/service-manual/helping-people-to-use-your-service/making-your-service-accessible-an-introduction (Accessed: 10 May 2024).

IMDb (2024a) *Advanced title search*, *IMDb*. Available at: https://www.imdb.com/search/ (Accessed: 10 May 2024).

IMDb (2024b) *IMDb - Can I use IMDb data in my software?* Available at: https://help.imdb.com/article/imdb/general-information/can-i-use-imdb-data-in-my-software/G5JTRESSHJBBHTGX?ref_=helpart_nav_18# (Accessed: 8 May 2024).

kenziekray (2023) ‘Best App for Tracking Movies/TV Shows (watched and want to watch)’, *r/ios*. Available at: www.reddit.com/r/ios/comments/126qcru/best_app_for_tracking_moviestv_shows_watched_and/ (Accessed: 10 May 2024).

Sommerville, I. (2015) *Software Engineering, Global Edition*. 10th edn. Harlow, UNITED KINGDOM: Pearson Education, Limited. Available at: http://ebookcentral.proquest.com/lib/surrey/detail.action?docID=5185655 (Accessed: 9 May 2024).

TMDB (2024a) *API Terms of Use — The Movie Database (TMDB)*. Available at: https://www.themoviedb.org/api-terms-of-use (Accessed: 8 May 2024).

TMDB (2024b) *TMDB’s API*, *The Movie Database (TMDB)*. Available at: https://developer.themoviedb.org/docs/faq (Accessed: 8 May 2024).

Trickyguy37 (2022) ‘BEST platform to keep track of the movies and series you have watched or plan to watch?’, *r/movies*. Available at: www.reddit.com/r/movies/comments/xpfrji/best_platform_to_keep_track_of_the_movies_and/ (Accessed: 10 May 2024).